Applied Ontology 0 (2019) 1–0
IOS Press

On the computational realization of Formal Ontologies: Formalizing an ontology of instantiation in spacetime using Isabelle/HOL as a case study

Thomas Bittner
Department of Philosophy, State University of New York at Buffalo, 135 Park Hall, Buffalo, NY, 14260.
e-mail: bittner3@buffalo.edu

Abstract. This paper shows in a case study that for the development, the presentation, and the computer-assisted verification of formal ontologies the usage of higher-order languages and associated proof assistant tools is highly beneficial. This case study demonstrates that the expressive power of a higher order logic in conjunction with a well developed infrastructure for theory development and presentation facilitate the development of formal ontologies in a way that is similar to the ways in which modern object oriented programming languages and associated IDEs facilitate the development of complex software. In particular ontology development in such an environment supports (a) the formal verification of the satisfaction of the axioms of a formal ontology in a class of structures that constitute its intended interpretation; (b) the computational instantiation of specific prototypical examples/models that guide the ontology development; and (c) the formal verification of proofs by demonstrating that the claimed theorems are indeed derivable from the axioms of the theory. Parallels to software development can be drawn for two reasons: Firstly, due to the non- or semi-decidability and the complexity of sufficiently expressive languages, the process of theory development, like software development, is computer assisted rather than fully automated. Secondly, the use of a higher order logic supports modularization, object orientation, model building and other features that that greatly simplify the development of complex formal ontologies.

Keywords: ontology of spacetime, modal logic, Isabelle/HOL/Isar

1. Introduction

Ontology is the study of what can possibly exist Lowe (2002) and Smith (2003). In *formal* ontology the axioms of formal theories are used to constrain logical possibilities to what is possible metaphysically. Unfortunately, developing formal theories is complex because it is difficult to determine all the consequences of even small sets of non-trivial axioms. Moreover, it is very easy to render a set of axioms inconsistent. Inconsistent sets of axioms are useless for formal ontology because they do not have any models and realizations at all.

The difficulties of the development of formal ontologies are well understood and have lead to the emergence of languages such as OWL (W3C OWL Working Group, 2012) with associated computational tools (Knublauch et al., 2004; Horrocks, 1998; Sirin et al., 2007; Haarslev and Möller, 2003) that (a) automatically ensure the consistency of sets of axioms and that (b) automatically derive the consequences of a given sets of axioms. It is also well understood that the expressive power of OWL-like languages is rather restricted and too limited for expressing relational aspects of formal ontology as well as aspects that go beyond what can be formulated as a classification problem (Bittner and Donnelly, 2007). Moreover, it is widely believed that the language of first order predicate logic is sufficiently powerful for formalizing formal ontologies (Smith, 2003). This may be true. The aim of this paper is it, however, to make the case that for the development, the presentation, and the computer-assisted verification of formal ontologies the usage of higher-order languages and associated tools is highly beneficial.

The usage of higher-order languages is beneficial because even in first order logic non-trivial theorem proving is only semi-decidable and highly complex (Lemon and Pratt, 1997; Loui, 1996; Renz and Nebel, 1999). Therefore, full proof automatization cannot be a goal and one has to settle for computational proof assistants that facilitate the semi-automated development of formal theories. In the realm of computer assisted theorem proofing the additional expressive power of higher order logics facilitate the development of an infrastructure that is highly beneficial for the development and the presentation of formal ontologies in ways that are similar to the ways in which object orientation is beneficial to software development (Bateman et al., 2007; Kammüller et al., 1999; Mossakowski et al., 2007). This paper illustrates these advantages by describing the development and the computational realization of a formal ontology that was originally presented semi-formally by Bittner (2018). The reasons for developing a computational realization of this formal ontology were threefold: (A) There was a need to formally verify the consistency of the formal theory, (B) There was a need to formally verify whether or not specific mathematical representations of physical systems are models of the formal theory, and (C) There was a need to formally verify that the theorems of the formal theory are derivable from a set of axioms that underwent multiple changes in the course of the development of the formal theory. Points (A) and (B) are important because many currently existing ontologies have a significant number of axioms the consistency of which needs to be verified and the models of which need to be explored. The ontology that is used here as an example has only 26 axioms. But already for a set of axioms of this size it is far from obvious whether or not the resulting theory is consistent. Moreover, even if the theory is is consistent, it is far from obvious whether or not the intended models are among the actual models of the formal theory. Point (C) is important because ontology development often is an iterative process and each iteration may see changes in the axioms of the formal ontology. In what follows an ontology will be called *formalized* if and only if there exists a computational realization of that ontology within which points (A) – (C) are computationally verified.

For the development and the computational realization of the formal ontology presented semi-formally by Bittner (2018), the HOL-based framework Isabelle/HOL/Isar (Paulson and Nipkow, 2017; Paulson, 1994; Nipkow, 2003) is used. HOL is a framework of higher order logic which combines predicate logic (Copi, 1979) with lambda calculus (Church, 1941) in a way that is based on Church's Theory of types (Church, 1940). Isabelle/Isar (Nipkow, 2003) is a language for writing formal theories within the logic HOL and which enables the 'programing' of structured and human-readable proofs. In some ways, writing a proof in Isabelle/Isar is like writing a function in an interpreted programming language. In addition the Isabelle/HOL/Isar framework provides a formal infrastructure – `locales` (Kammüller et al., 1999; Ballarin, 2004) – which allow to incorporate features of object-orientation in the development of formal theories. More technically `locales` in Isabelle/HOL are generalizations of axiomatic type classes (Jones and Jones, 1997; Wenzel, 2005) that originally were introduced as part of the functional language Haskell (Thompson, 1999). Similar to axiomatic type classes, `locales` in Isabelle/HOL provide a formal infrastructure for (a) modular theory development by maintaining hierarchical links between theories (Tab. 2 on pg. 31) and (b) the establishment of formal links between theories and the structures that are models of such theories. In what follows `locales` are used heavily to achieve the goals (A) – (C) set out above.

From a more methodological perspective, the computational realization of a formal ontology in Isabelle/HOL/Isar facilitates the explicit separation of three fundamental levels of ontology development: (I) the level of axiomatization; (II) the level of model instantiation; and (III) the level of theory presentation.

The level of theory presentation: Roughly, the level of theory presentation corresponds to the way a formal ontology is usually presented semi-formally in a scientific publication. The formal ontology considered in this case study was originally presented under the title "Formal ontology of space, time and physical entities in modern classical mechanics" (Bittner, 2018). The aim of this ontology was to distinguish logical possibilities from metaphysical possibilities from various kinds of physical possibilities in the context of a formal theory that has two major primitives: the parthood relation among regions of spacetime and the relation of instantiation of universals by physical entities at regions of spacetime.

For the semi-formal presentation of this formal ontology a modal predicate logic was used. The modal operators provide means to talk about physical possibility/necessity as well as means to talk about aspects of the underlying spacetime structures that have different descriptions in different frames of reference. Using a modal language facilitates conceptual clarity while maintaining formal rigor. The choice of a modal language greatly simplifies the presentation by focussing on conceptual and logical issues and by hiding the specifics of the underlying interpretation. In contrast to the semi-formal presentation of the formal theory by Bittner (2018), in the computational realization discussed below the axioms of Bittner (2018) are stated 'semantically' (at the level of axiomatization) and are then 'lifted' to the modal language using an encoding of modal logic into Isabelle/HOL (or any other HOL framework) that was developed originally by Benzmüller (2015) and Benzmüller and Woltzenlogel Paleo (2015).

The level of axiomatization: At the level of axiomatization the axioms of Bittner (2018) are expressed in a non-modal language with explicit reference to the structures in which they are interpreted. The latter include the sets that determine the domains of quantification, the accessibility relations, as well as the relations that serve as the interpretation of the primitives of the formal theory. In essence, in the computational representation the axiomatization is realized in a non-modal second-order language at a level that corresponds to the level of interpretation in the presentation of the formal theory of Bittner (2018). Semantically, this constitutes a deep embedding of this formal theory into Isabelle/HOL. Similar techniques have been used for example in the work by Foster et al. (2015).

The level of model instantiation: The `locale` constructs of the Isabelle/HOL system provide the infrastructure for instantiating abstract model-theoretic structures associated with a set of axioms by specific models and for creating proof obligations that, when fulfilled, ensure that the specific model satisfies all the axioms that are associated with the abstract model-theoretic structures. Roughly, if a model finder verifies that a set of axioms is consistent then it confirms that the class of structure associated with the set of axioms is not empty. The instantiation of a specific model – if selected appropriately – verifies that the class of *intended* models is not empty. The infrastructure provided by Isabelle/HOL and its `locales` thereby provides formal means to verify the consistency of a set of axioms as well as means to verify that the intended models are among the actual models of a formal theory. While in the semi-formal presentation one has to talk about an intended interpretation, in the computational realization one can talk about *enforced* interpretations.

The remainder of this paper is structured as follows. Since the formal ontology presented by Bittner (2018) serves as the running example of a theory which semi-formal presentation is to be complemented by a rigorous computational realization, this example ontology needs to be explained at least briefly. In this presentation the theory is taken 'as is' and no attempts are made to justify any of the choices, commitments and presuppositions that underly the formal development. For a discussion of those aspects please consult the original paper. To maintain a reasonable degree of self-containment, the differential geometry of spacetime that provides the framework in which the formal models are specified is briefly summarized in the appendix (pg. 33). After the brief discussion of the background that motivated the development of the example theory, the levels (I)–(III) of its computational realization are discussed in detail. The fully formalized and computationally verified formal ontology can be found at: `http://www.buffalo.edu/~bittner3/Theories/OntologyCM/` and `http://www.buffalo.edu/~bittner3/Theories/OntologyCM/sources/`.

2. Formalizing physical and metaphysical possibilities

Fundamental to ontologies of dynamical phenomena is it to formally distinguish the following classes of sequences of changes and corresponding processes: (i) changes and processes that are logically and combinatorially possible; (ii) changes and processes that are *meta*physically possible; (iii) changes and processes that are *physically* possible. For example, instantaneous changes are logically and metaphysically possible for immaterial entities (e.g., fiat boundaries (Smith and Varzi, 2000)) but physically impossible for material entities. To formally distinguish logical and metaphysical possibilities from physical possibilities in the ontology of Bittner (2018) a modal logic of parthood, instantiation and location was proposed. In this formal ontology the modal operators are used to express what, according to modern classical mechanics, is true on some physical possibility and to distinguish it from what is true, again, according to modern classical mechanics, on all physical possibilities.

Bittner (2018) characterizes physical possibilities along two 'dimensions': *physically possible worldlines* and *physically possible slicings of spacetime* into hyperplanes of simultaneity. Physically possible worldlines are regions of spacetime that can be occupied by physically possible processes and along which physically possible continuants can evolve by realizing physically possible sequences of states. Physical constrains on possible worldlines also restrict the ways in which complex systems can arise from the possibilities of simple systems and thereby affect the mereological structure of physically possible entities.

The second 'dimension' of characterizing physical possibilities by Bittner (2018) is concerned with possible frames of reference and the slicings of spacetime into hyperplanes of simultaneity those frames impose. As pointed out in the theory of Special Relativity (Einstein, 1951), it is meaningless to speak of space and time without reference to a specific slicing of spacetime into hyperplanes of simultaneity. Moreover, only within a given slicing of spacetime it makes sense to relate physically possible worldlines to constraints on the causal structure of the physical world. Similarly, only within a given slicing of spacetime it is meaningful to speak about metaphysical (e.g., mereological, topological, etc.) constraints on continuant entities and the changes they can possibly undergo. Thus, it is ontologically relevant to distinguish what is true on merely some slicings of spacetime from what is true under all possible slicings of spacetime.

At the formal level the two 'dimensions' of physical possibilities find their expression in a two-dimensional modal logic. The class of structures in which this modal language is interpreted ($\mathcal{KS}$-structures) is discussed in the next subsection. A brief overview of the differential geometry that is used to describe physically possible worldlines and the slicing of spacetime into hyperplanes of simultaneity in $\mathcal{KS}$-structures is given in the appendix. Tab. 1 summarizes some important notions discussed there. For more details see the original presentation of Bittner (2018). The Syntax and the semantics of the modal language are introduced in Sec. 2.3.

2.1. $\mathcal{KS}$-structures

In the ontology presented by Bittner (2018) what is physically possible according to the constraints imposed by classical mechanics is encoded in set-theoretic structures that give rise to $\mathcal{KS}$-structures of the form

$$\mathcal{KS}(m,\mathcal{L}) =_{df} \langle \mathcal{D}_{ST}, \mathcal{D}_E, \mathcal{K}, \mathsf{V}, \underline{\sqcup}, \overline{\sqcap}, \mathsf{TS}, \mathsf{InstST}, \mathsf{AtE} \rangle. \tag{1}$$

The parameters m and $\mathcal{L}$ respectively specify the number of atomic particles and the Lagrangian field that constrains the physically possible changes a world with m atomic particles can undergo. Both parameters are determined empirically (Appendix C).

The sets $\mathcal{D}_{ST}$ and $\mathcal{D}_E$ of $\mathcal{KS}(m,\mathcal{L})$ are respectively the domains of regions of spacetime (sub-manifolds of the spacetime manifold) that can possibly exist and the domain of entities that can possibly exist. $\mathcal{K}$ is a modal frame structure on the set of physical possibilities. V is the interpretation function. The sets $\underline{\sqcup}$, $\overline{\sqcap}$, TS, InstST, AtE of $\mathcal{KS}(m,\mathcal{L})$ serve as the interpretations of the axiomatic primitives of the formal theory in the context of the physical possibilities in $\mathcal{K}$. In the remainder of this subsection the focus is on $\mathcal{K}$, $\mathcal{D}_{ST}$ and $\mathcal{D}_E$. The interpretation function V is discussed when the syntax and the semantics of the formal language are introduced in Sec. 2.3. The sets $\underline{\sqcup}$, $\overline{\sqcap}$, TS, InstST and AtE are introduced as the computational representation of the formal ontology is developed.

Regions of spacetime and physical possibilities: The members of $\mathcal{D}_{ST}$ are the regions of spacetime (sub-manifolds of the spacetime manifold in the sense of A.1). In particular $\mathcal{D}_{ST}$ includes spacetime itself – a manifold of topology $\mathcal{ST} = (\Re \times M)$, i.e., $\mathcal{ST} \in \mathcal{D}_{ST}$. $\mathcal{D}_{ST}$ also includes the set the members of Γ – the set of geometrically possible worldlines. Those are curves through spacetime along which processes that involve systems that are constituted of up to m particles and a Lagrangian field $\mathcal{L}$ can possibly evolve according to the laws of classical mechanics. (See also A.2, B.1.)

symbolic expression	description	Appendix
M	Manifold	A.1
$M_1 \sqsubseteq M_2$	M_1 is a submanifold of M_2	A.1
$\bigsqcup S$	Join (union) of a non-empty set S of manifolds such that the result is a manifold.	A.1
$\bigsqcap S$	Meet (intersection) of a non-empty set set S of manifolds such that the result is a manifold.	A.1
T_xM	Tangent space on manifold M at point $x \in M$	A.1
TM	Tangent bundle on M. TM is the disjoint union of the tangent spaces T_xM for all $x \in M$	A.1
$\gamma : \Re \rightarrow M$	parametric curve on M	A.2
$\gamma \sqsubset M$	$\gamma =_{df} \{\gamma(\tau) \in M \mid \tau \in \Re\}$ is the curve $\gamma \subset M$ represented by the parametric curve $\gamma(\tau)$ with $\tau \in \Re$	A.2
$(\mathcal{ST}, g)$	Spacetime manifold of topology $(\Re \times M)$ and geometry (metric field) g	B.1, B.2
g_x	Metric field: $g_x : T_x\mathcal{ST} \times T_x\mathcal{ST} \rightarrow \Re$ at $x \in \mathcal{ST}$, defines the length $\lvert\xi\rvert$ of a vector $\xi \in T_x\mathcal{ST}$ via $\lvert\xi\rvert^2 = g_x(\xi, \xi)$	B.1, B.2
$(\mathcal{T}, g_\mathcal{T})$	Abstract time slice with geometry $g_\mathcal{T}$. If $\mathcal{ST}$ has the topology $\Re \times M$ then $\mathcal{T}$ has the dimension of M	B.1
$\mathcal{T}$-slicing σ	A $\mathcal{T}$-slicing σ of $(\mathcal{ST}, g)$ is a smooth map $\sigma : \Re \times \mathcal{T} \rightarrow (\Re \times M)$	B.1
$\sigma_t(\mathcal{T})$	Concrete timeslice (time instant, hyperplane of simultaneity) of spacetime according to the $\mathcal{T}$-slicing σ. I.e., $\sigma_t(\mathcal{T}) = \{(t, \sigma_t(x)) \mid x \in \mathcal{T}\}$. σ_t is a isomorphism from $\mathcal{T}$ to $\sigma_t(\mathcal{T})$	B.1
$\sigma(\mathcal{T})$	A particular slicing of spacetime $\mathcal{ST}$ into hyperplanes of simultaneity	B.1
Γ	The set of geometrically possible worldlines	B.1
Σ	The set of $\mathcal{T}$-slicings of a given underlying spacetime	B.1
$\xi = \frac{d}{d\tau}\gamma(\tau)\lvert_x$	ξ is the tangent on γ at point $x \in \gamma$	A.2
$H : M \rightarrow \Re$	H is a scalar field on M	C
$X : M \rightarrow TM$	X is a vector field on M such that $X(x) \in T_xM$ for all $x \in M$	C
$\gamma_{X,x} : \Re \rightarrow M$	integral curve through $x \in M$ with respect to the vector field $X : M \rightarrow TM$. I.e., if $\gamma(\tau) = y$ then $X(y)$ is the tangent on γ at $y \in M$	C
$\mathcal{L} : T\mathcal{ST} \rightarrow \Re$	The Lagrangian field: a scalar field on the tangent bundle of the spacetime manifold. Determines physically possible worldlines.	C
$\Gamma^\mathcal{L}$	The set of physically possible worldlines as determined by $\mathcal{L}$.	C

Table 1

Summary the Appendix (pg. 33): Basic notions of differential geometry (Arnold, 1997; Butterfield, 2007; Bittner, 2018)

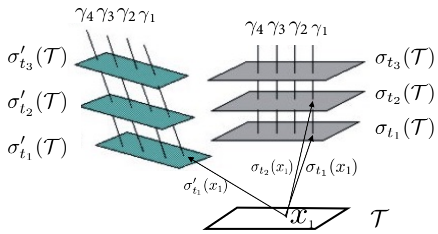
Σ is a set of $\mathcal{T}$-slicings σ of spacetime. A $\mathcal{T}$-slicing of spacetime is a smooth mapping $\sigma : \Re \times \mathcal{T} \rightarrow (\Re \times M)$ such that for every instant $t \in \Re$ of time, σ_t maps the points of an abstract n-dimensional Euclidean space, $\mathcal{T}$, to the points of an n-dimensional slice $\sigma_\mathrm{t}(\mathcal{T}) \subseteq \mathcal{ST}$ of the $n+1$-dimensional spacetime manifold $\mathcal{ST}$ (see also B.1). For all slicings $\sigma \in \Sigma$, the set of all time slices $\sigma_\mathrm{t}(\mathcal{T})$ is a subset of $\mathcal{D}_{ST}$. Spatial regions are members of $\mathcal{D}_{ST}$ that emerge from the intersection of worldlines and time slices. That is, if $\gamma \in \Gamma$ and $\gamma \cap \sigma_\mathrm{t}(\mathcal{T}) \neq \emptyset$ then $\gamma \cap \sigma_\mathrm{t}(\mathcal{T}) \in \mathcal{D}_{ST}$. A subset of the geometrically possible worldlines in Γ is physically possible, i.e., $\Gamma^\mathcal{L} \subseteq \Gamma$. That is, $\Gamma^\mathcal{L}$ is a set of physically possible worldlines along which worlds/systems with m particles can evolve according to the Lagrangian $\mathcal{L}$. (See C.)

The product $\Gamma^\mathcal{L} \times \Sigma$ is the set of all physical possibilities. In conjunction with two accessibility relations R^Γ and R^Σ the set of physical possibilities forms a frame structure $\mathcal{K} = (\Gamma^\mathcal{L} \times \Sigma)$. The accessibility relations R^Γ and R^Σ of the resulting frame structure are axiomatized as:

$$\begin{array}{l} R^\Gamma \subseteq \{\langle\langle\gamma_1, \sigma\rangle, \langle\gamma_2, \sigma\rangle\rangle \mid \langle\gamma_1, \sigma\rangle, \langle\gamma_2, \sigma\rangle \in \mathcal{K}\} \text{ and } R^\Gamma \text{ is reflexive, symmetric, and transitive;} \\ R^\Sigma \subseteq \{\langle\langle\gamma, \sigma_1\rangle, \langle\gamma, \sigma_2\rangle\rangle \mid \langle\gamma, \sigma_1\rangle, \langle\gamma, \sigma_2\rangle \in \mathcal{K}\} \text{ and } R^\Sigma \text{ is reflexive, symmetric, and transitive.} \end{array} \tag{2}$$

That is, both, $(\Gamma^\mathcal{L}, R^\Gamma)$ and (Σ, R^Σ) are structures with an equivalence relation. In addition the two accessibility relations are compositionally related as indicated in (Fig. 1 (right)). For a justification of those choices see the original presentation of Bittner (2018).

Example 1. Consider the left image of Fig. 1. It displays a three-dimensional spacetime with two spatial dimensions and one temporal dimension. There are the geometrically possible ('straight') worldlines $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\} \subseteq \Gamma$. Suppose that other curvy but smooth and monotonically increasing worldlines are



$$\sigma^1 \xrightarrow{R^\Sigma} \sigma^2$$

$\mathcal{F}_\Sigma$

$$\begin{tikzcd} \gamma^2 \\ \gamma^1 \arrow[u, "R^\Gamma"] \end{tikzcd}$$

$\mathcal{F}_\Gamma$

$$\begin{tikzcd} \langle \gamma^2, \sigma^1 \rangle \arrow[r, "R^\Sigma"] & \langle \gamma^2, \sigma^2 \rangle \\ \langle \gamma^1, \sigma^1 \rangle \arrow[u, "R^\Gamma"] \arrow[r, "R^\Sigma"] & \langle \gamma^1, \sigma^2 \rangle \arrow[u, "R^\Gamma"'] \end{tikzcd}$$

$\mathcal{F}_\Sigma \times \mathcal{F}_\Gamma$

Fig. 1. $\mathcal{T}$-slicings σ' and σ of a spacetime with worldlines $\gamma_1 - \gamma_4$ (left); Two accessibility relations in distinct frames and two accessibility relations in a two dimensional product frame (right) (adapted from the work of Gabbay (2003, pg. 125))

also geometrically possible. In the image there are two slicings σ and σ' of spacetime, i.e., $\Sigma = \{\sigma, \sigma'\}$. The abstract time slice is $\mathcal{T} = (\Re^2, g)$ and g is the Euclidean metric of $\Re^2$. Suppose that the world is such that it has one particle ($m = 1$) which, in addition is such that it cannot change its spatial location, i.e., the Lagrangian field $\mathcal{L}$ 'holds' the particle in place as time passes. In such a world are then four physically possible worldlines, i.e., $\Gamma^{\mathcal{L}} = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$. The physical possibilities are the members of the set $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\} \times \{\sigma, \sigma'\}$. In this world the slicing σ corresponds to the particle's rest frame and the slicing σ' corresponds to a frame of reference that is in motion relative to the particle. This gives rise to the product frame $\mathcal{K}(\Gamma^{\mathcal{L}}, \Sigma)$ in which the relations R^Γ and R^Σ hold: $\gamma_1 R^\Gamma \gamma_1$, $\gamma_1 R^\Gamma \gamma_2$, $\gamma_2 R^\Gamma \gamma_3$, $\gamma_1 R^\Gamma \gamma_3$, $\ldots$, $\sigma R^\Sigma \sigma$, $\sigma R^\Sigma \sigma'$, etc. $\Box$

The domain of entities: On the enforced interpretation $\mathcal{D}_E$ is the domain of possible entities (particulars and universals) in a world with m atoms. While the number and kinds of atomic particles that exist are fixed, whether and which complex continuants are formed by the given atomic entities is a contingent matter. Whatever complex entities can exist, however, must obey the laws of mereology in a way that is consistent with the mereology of the underlying spacetime. The domain $\mathcal{D}_E$ of possible entities and the domain $\mathcal{D}_{ST}$ of regions of spacetime are linked via the relation of instantiation $\mathsf{InstST} \subseteq \mathcal{D}_E \times \mathcal{D}_E \times \mathcal{D}_{ST} \times \mathcal{K}$. More details will be discussed as the computational representation of the formal ontology is developed.

2.2. A simplified model

For illustrative purposes and to check the consistency of the formal theory it will be useful to use a simple and finite set-theoretic model to illustrate some important aspects of the class of $\mathcal{KS}$-structures. Due to its simplistic nature this model falls short of capturing many of the topological, geometric and differential structures that govern the underlying physics. More sophisticated models could be built by implementing Def. 1 of appendix B.1 and basing it on manifold theory and the theory of differential forms (Arfken et al., 2005). An important advantage of using a tool with the expressive power of Isabelle/HOL is that, at least in principle, it is possible to formalize models of this kind.

The toy model has a two-dimensional 'spacetime'. This spacetime is discrete and has six distinct spatio-temporal locations as indicated in Fig. 2(a), i.e., $\mathcal{ST} = \{c_{00}, c_{10}, c_{01}, c_{11}, c_{02}, c_{12}\}$ as indicated by the labeling in the figure. In this spacetime the domain of spacetime regions is the set of non-empty subsets of $\mathcal{ST}$, i.e., $\mathcal{D}_{ST} = \{r \subseteq \mathcal{ST} \mid r \neq \emptyset\}$. As discussed above (Fig. 1), slicings of spacetime are mappings σ^i from an abstract timeslice $\mathcal{T}$ of dimension $n-1$ into into an n dimensional spacetime $\mathcal{ST}$. In this toy model the abstract time slice is $\mathcal{T} = (\{x_0, x_1\}, g)$. The geometry g of $\mathcal{T}$ is mostly ignored here. In the model there are two slicings of spacetime $\Sigma = \{\sigma^0, \sigma^1\}$. The slicing σ^0 is $\sigma^0_0(x_i) = c_{i0}$, $\sigma^0_1(x_i) = c_{i1}$, $\sigma^0_2(x_i) = c_{i2}$ for $i \in \{0, 1\}$ (Fig. 2(b)) and the slicing σ^1 is $\sigma^1_0(x_i) = c_{10}$, $\sigma^1_1(x_0) = c_{00}$, $\sigma^1_1(x_1) = c_{11}$, $\sigma^1_2(x_0) = c_{01}$, $\sigma^1_2(x_1) = c_{12}$, $\sigma^1_3(x_i) = c_{02}$ for $i \in \{0, 1\}$ (Fig. 2(c)). (Clearly, unlike σ^0, σ^1 is not an isomorphism that preserves the geometry g. This is an artifact of the finite nature of $\mathcal{ST}$ in this model.)

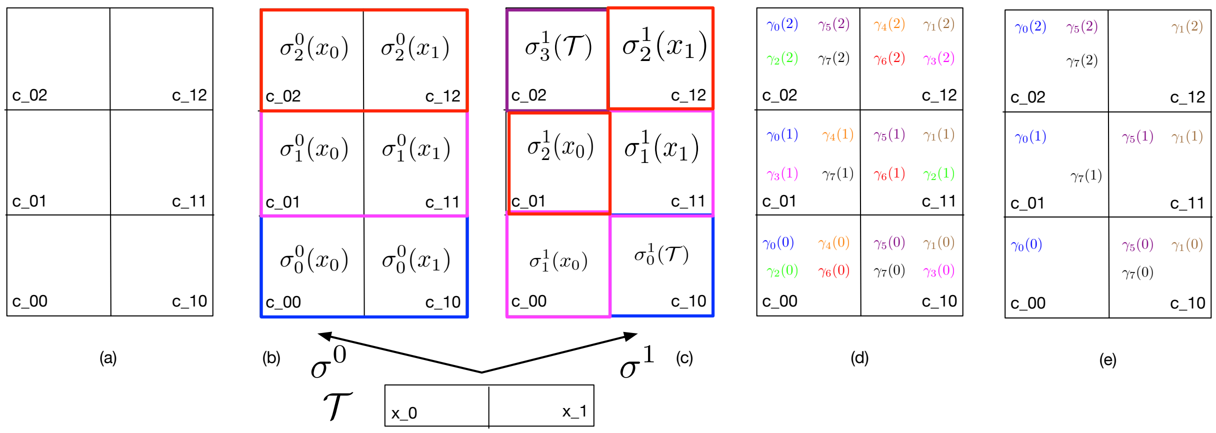


Fig. 2. A simple toy model of spacetime in conjunction with examples for physically possible worldlines and timeslices. Color coding: (b) $\sigma_0^0(\mathcal{T})$ is blue, $\sigma_1^0(\mathcal{T})$ is magenta and $\sigma_2^0(\mathcal{T})$ is red; (c) $\sigma_0^1(\mathcal{T})$ is blue, $\sigma_1^1(\mathcal{T})$ is magenta, $\sigma_2^1(\mathcal{T})$ is red, $\sigma_2^1(\mathcal{T})$ is black; worldlines in (d) and/or (e) γ_0 is blue, γ_1 is brown, γ_2 is green, γ_4 is orange, γ_5 is purple, γ_6 is red, γ_7 is black.

The worldlines that are kinematically possible with respect to the slicing σ^0 are visualized in Fig. 2(d). The worldlines that are kinematically possible with respect to the slicings σ^0 and σ^1 are visualized in Fig. 2(e) and listed in Eq. 3. There are fewer worldlines in Fig 2(e) because a worldline cannot have more than one 'point' of intersection with a time slice. This is because instantaneous changes are physically impossible.

The parameter of the γ_i is understood to correspond to the (coordinate) time (Def. 2 of B.1) according to the slicing σ^0 of $\mathcal{ST}$, i.e., $\tau \in 0 \ldots 2$. (The aim here is to approximate worldlines that are possible in a spacetime that is consistent with the special theory of Relativity – See also B.2.)

$$\begin{array}{ll} \gamma_0(\tau) = c_{0\tau}, & \gamma_5(0) = c_{10}, \gamma_5(1) = c_{11}, \gamma_5(2) = c_{02}, \\ \gamma_1(\tau) = c_{1\tau}, & \gamma_7(0) = c_{10}, \gamma_7(1) = c_{01}, \gamma_7(2) = c_{02}. \end{array} \tag{3}$$

If one demands, in accordance with classical mechanics, that distinct particles cannot occupy the same location in spacetime then worldlines of distinct particles cannot intersect. In this example it is assumed that there exist two atomic particles that occupy locations along the kinematically possible worldlines in this spacetime. The worldlines γ_0 and γ_1 are the only kinematically possible particle worldlines that do not intersect. The only kinematically possible complex worldline in this example is $\gamma_0^2 = \bigcup\{\gamma_0, \gamma_1\}$. (The underlying physical environment that is encoded in the Lagrangian field $\mathcal{L}$ would be such that neither of the two atomic particles can change its spatial location.) On these assumptions, $\mathcal{K} = \Gamma^{\mathcal{L}} \times \Sigma$, the set of physical possibilities, is $\mathcal{K} = \{\gamma_0^2\} \times \{\sigma^0, \sigma^1\}$.

In what follows the two atomic particles are called At_0 and At_1. Respectively At_0 and At_1 evolve along the worldlines γ_0 and γ_1. In addition it is assumed that there exists a complex object Compl_0 that is constituted by the atoms At_0 and At_1. The worldline of Compl_0 is γ_0^2. Within the realm of physical possibilities in $\mathcal{K}$ an ontology that commits to the existence of continuant particulars, occurrent particulars as well as to universals which are instantiated by such particular entities (e.g., BFO (Smith, 2016), DOLCE (Gangemi et al., 2003), etc.) then is committed to acknowledging the existence of at least the following entities: the continuants At_0, At_1, and Compl_0; the occurrents Occ_0, Occ_1, and Occ_2 (the respective lives of the above continuants); and at least two universals (UC_0 and UO_0) which are respectively instantiated by the continuants and occurrents. On the given assumptions the set of physically possible entities is: $\mathcal{D}_E = \{\mathsf{At}_0, \mathsf{At}_1, \mathsf{Compl}_0, \mathsf{Occ}_0, \mathsf{Occ}_1, \mathsf{Occ}_2, \mathsf{UC}_0, \mathsf{UO}_0\}$.

This example is constructed to minimize the number of physical possibilities without being trivial. This simplicity of the example model greatly reduces the complexity of the (mostly brute force and case-based) proofs that establish that the axioms of the formal ontology are satisfied in this toy model. In general, within the framework of highly expressive languages, the more skilled the developer of the computational realization of an ontology, the more realistic and sophisticated the models that are realized can be. In the conclusions of this paper I will argue for the need of skilled proof engineers.

2.3. Semi-formal specification of the modal language

In this section the syntax and the semantics of the formal language that is used to express the ontology of Bittner (2018) is introduced. This language includes three disjoint sets of variable symbols: Var_{ST}, VAR_{ST}, and Var_E. Var_{ST} contains variables denoted by letters u, v, w, possibly with subscripts (u_1, v_2, etc.). VAR_{ST} contains variables denoted by capital letters A, B, etc. Var_E contains variables x, y, z, possibly with subscripts. *Var* is the union $\mathit{Var}_{ST} \cup \mathit{VAR}_{ST} \cup \mathit{Var}_E$. The sets $\mathcal{D}_{ST}$, $\mathcal{P}(\mathcal{D}_{ST})$ and $\mathcal{D}_E$ of Sec. 2.1 are respectively the domains for the variables in Var_{ST}, VAR_{ST}, and Var_E.

Pred is a set of predicate symbols. If F is an n-ary predicate symbol in *Pred* and $t_1, \ldots, t_n$ are variables in *Var* then $F\ t_1 \ldots t_n$ is a well-formed formula. Complex, non-modal formulas are formed inductively in the usual ways, i.e., if α and β are well-formed formulas, then so are $\neg\alpha, \alpha \wedge \beta, \alpha \vee \beta, \alpha \rightarrow \beta, (x)\alpha, (\exists x)\alpha$ (Gabbay, 2003; Hughes and Cresswell, 2004). All quantification is restricted to a single sort of variables. If not marked explicitly, restrictions on quantification are understood by conventions on variable usage. Finally, the modalities $\Box^\Gamma$, $\Box^\Sigma$, $\Diamond^\Gamma$ and $\Diamond^\Sigma$ are included in the formal language, i.e., if α is a well-formed formula, then so are $\Box^i\alpha$ and $\Diamond^i\alpha$ with $i \in \{\Gamma, \Sigma\}$.

A model of such a multi-dimensional sorted modal language is a structure $\langle \mathcal{D}_{ST}, \mathcal{D}_E, \mathcal{K}, \mathsf{V}\rangle$. $\mathcal{D}_{ST}$ and $\mathcal{D}_E$ are as described above and form the non-empty domains of quantification. $\mathcal{K}$ is a non-empty set of possible worlds, which, as discussed above, has the internal structure of a product of two sets $\Gamma^{\mathcal{L}}$ and Σ. $\mathcal{K}$ gives rise to the product frame of the two-dimensional modal logic (Gabbay, 2003) presented here. V is the interpretation function: if $F \in \mathit{Pred}$ is an n-ary predicate then $\mathsf{V}(F)$ is a set of $n+1$-tuples of the form $(\mathsf{d}_1, \ldots, \mathsf{d}_n, \kappa)$ with $\mathsf{d}_1, \ldots, \mathsf{d}_n \in \mathcal{D}$ and $\kappa \in \mathcal{K}$, where $\mathcal{D} = \mathcal{P}(\mathcal{D}_{ST}) \cup \mathcal{D}_{ST} \cup \mathcal{D}_E$. In all possible worlds $\kappa \in \mathcal{K}$ the variables respectively range over all the members of $\mathcal{D}_{ST}$, $\mathcal{P}(\mathcal{D}_{ST})$ and $\mathcal{D}_E$. A variable assignment μ is a function such that (i) for every variable $u \in \mathit{Var}_{ST}$, $\mu(u) \in \mathcal{D}_{ST}$, (ii) for every variable $x \in \mathit{Var}_E$, $\mu(x) \in \mathcal{D}_E$, and (iii) for every variable $A \in \mathit{VAR}_{ST}$, $\mu(A) \in \mathcal{P}(\mathcal{D}_{ST})$.

Every well-formed formula has a truth value which is defined as follows:

$$
\begin{array}{ll}
0 & \mathsf{V}_\mu(F\, t_1 \ldots t_n, \kappa) = 1 \text{ if } \langle \mu(t_1), \ldots, \mu(t_n), \kappa\rangle \in \mathsf{V}(F) \text{ and 0 otherwise;} \\
1 & \mathsf{V}_\mu(\neg\alpha, \kappa) = 1 \text{ if } \mathsf{V}_\mu(\alpha, \kappa) = 0 \text{ and 0 otherwise;} \\
2 & \mathsf{V}_\mu(\alpha \wedge \beta, \kappa) = 1 \text{ if } \mathsf{V}_\mu(\alpha, \kappa) = 1 \text{ and } \mathsf{V}_\mu(\beta, \kappa) = 1 \text{ and 0 otherwise;} \\
3 & \mathsf{V}_\mu(\alpha \rightarrow \beta, \kappa) = 1 \text{ if } \mathsf{V}_\mu(\alpha, \kappa) = 0 \text{ or } \mathsf{V}_\mu(\beta, \kappa) = 1 \text{ and 0 otherwise;} \\
4 & \mathsf{V}_\mu(\Box^\Gamma\alpha, \kappa) = 1 \text{ if } \mathsf{V}_\mu(\alpha, \kappa') = 1 \text{ for all } \kappa' \in \mathcal{K} \text{ such that } R^\Gamma(\kappa, \kappa') \text{ and 0 otherwise,} \\
& \qquad \text{where } R^\Gamma \text{ is the accessibility relation on } \mathcal{K} \text{ for } \Box^\Gamma; \\
5 & \mathsf{V}_\mu(\Box^\Sigma\alpha, \kappa) = 1 \text{ if } \mathsf{V}_\mu(\alpha, \kappa') = 1 \text{ for all } \kappa' \in \mathcal{K} \text{ such that } R^\Sigma(\kappa, \kappa') \text{ and 0 otherwise,} \\
& \qquad \text{where } R^\Sigma \text{ is the accessibility relation on } \mathcal{K} \text{ for } \Box^\Sigma; \\
6 & \mathsf{V}_\mu((t)\alpha, \kappa) = 1 \text{ if } \mathsf{V}_\rho(\alpha, \kappa) = 1 \text{ for every } t\text{-alternative } \rho \text{ of } \mu \text{ and 0 otherwise,} \\
& \qquad \text{where a } t\text{-alternative } \rho \text{ of } \mu \text{ is a variable assignment that assigns the same domain} \\
& \qquad \text{members to all variables except for } t.
\end{array}
\tag{4}
$$

A well-formed formula α is true in $\langle \mathcal{D}_{ST}, \mathcal{D}_E, \mathcal{K}, \mathsf{V}\rangle$, i.e. $\mathsf{V}_\mu(\alpha) = 1$, if and only if $\mathsf{V}_\mu(\alpha, \kappa) = 1$ for all $\kappa \in \mathcal{K}$ and all assignments μ. Formula α is valid if α is true in all models. To simplify the presentation, the explicit distinction between V and V_μ will be omitted. Variables in the object language are written in *italics* and for corresponding domain members the Sans Serif font is used.

The formal theory includes the rules and axioms of a first order modal predicate logic with identity (Hughes and Cresswell, 2004) as well as the S5-axiom schemata $K_{\Box^i}$, $T_{\Box^i}$, and $5_{\Box^i}$ for $i \in \{\Gamma, \Sigma\}$. $\Diamond^i$ is defined in the usual way as the dual of $\Box^i$ for $i \in \{\Gamma, \Sigma\}$ ($D_{\Diamond^i}$). The Barcan formula and its converse are true in all models ($BC_{\Box^i}$).

$$
\begin{array}{llll}
D_{\Diamond^i} & \Diamond^i\alpha \equiv \neg\Box^i\neg\alpha & K_{\Box^i} & \Box^i(\alpha \rightarrow \beta) \rightarrow (\Box^i\alpha \rightarrow \Box^i\beta) \\
T_{\Box^i} & \Box^i\alpha \rightarrow \alpha & BC_{\Box^i} & (x)\Box^i\alpha \leftrightarrow \Box^i(x)\alpha \\
5_{\Box^i} & \Diamond^i\alpha \rightarrow \Box^i\Diamond^i\alpha & MS_{\Box} & \Box^\Gamma\Box^\Sigma\alpha \leftrightarrow \Box^\Sigma\Box^\Gamma\alpha
\end{array}
$$

Both modal operators are independent and the order of their application is immaterial ($MS_\Box$). All axioms of the formal theory below are true in all possible worlds have an implicit leading $\Box$ operator. In addition, leading universal quantifiers are omitted. Axioms $BC_{\Box^i}$ and $MS_\Box$ ensure that the order of leading universal quantifiers and leading $\Box$ operators is immaterial.

3. Computational realization of the formal language

The computational realization of the formal ontology starts with the computational realization of the semi-formal specification of the syntax and the semantics of the language of Sec. 2.3 in conjunction with the computational realization of the $\mathcal{KS}$-structures of Sec. 2.1. As mentioned above, HOL combines predicate logic with typed lambda calculus which results in a typed second order language. The formal roots in typed lambda calculus makes a typed functional language – in this case ML (Milner et al., 1990) – a natural choice for the implementation of a framework such as Isabelle/HOL. From the underlying ML system Isabelle/HOL inherits the basic syntax, the strong typing system, and the capability to evaluate functions. On top of the ML system, Isabelle/HOL then provides a derivability relation (`==>`) between objects of type (list of) formula (Nipkow et al., 2002). This in turn is the foundation for a number of object logics (Paulson, 1995) – among them the logic HOL. Within HOL then the datatype (`'a set`) is declared which stands for "set of type `'a`" where `'a` is a type variable which can be instantiated by specific datatypes (like `bool` or `int`). Within this framework then from existing sets new sets can be constructed via restricted (typed) set comprehension, Cartesian products, etc. In what follows the `typewriter font` is used for expressions of the Isabelle/HOL/Isar framework.

3.1. Representing product frames

Consider Fig. 3. The depicted Isabelle/HOL/Isar code illustrates the declaration of the computational representation of a product frame $\mathcal{K} = (\Gamma^{\mathcal{L}} \times \Sigma)$ of the form described in Fig. 1 (right) and Eq. 2. In lines 1–5 a record type `('a RS_frame)` is declared which instances are ordered quadtuples of the form (`r_carrier`, `aR`, `s_carrier`, `aS`). In this declaration `r_carrier` stands for a variable of the type "set of sets with members of type `('a)`", declared by the expression `r_carrier :: ('a set) set`. Respectively, `aR` stands for a variable of the type "function of type ((`'a set`) $\times$ (`'a set`) $\rightarrow$ `bool`)" – the computational representation of a binary relation with arguments of type (`'a set`). The types of `s_carrier` and `aS` are declared in analogy to the types of `r_carrier` and `aR`. The expression `(infixl "R₁" 50)` declares that the binary relation `aR` is abbreviated as `R` and the arguments are written in infix notation. The number 50 specifies the strength of the binding to minimize the number of parenthesises that are needed and thereby to facilitate readability. The subscript $_1$ specifies that the relation `R` has an argument which is the record structure itself. This argument is written as a subscript can be omitted in many situations. (This is somewhat similar to the *this* or *self* pointers in languages such as C^{++} or Python.) Similarly to `aR` and $\mathtt{R}_1$, `aS` has the infix notation $\mathtt{S}_1$.

Locales in Isabelle/HOL/Isar provide an infrastructure that supports the assignment of sets of axioms to types of structures. (This is similar to axiomatic type classes in HASKELL (Jones, 1993; Wenzel, 2005).) Consider the lines 6 – 19 of Fig. 3. Line 6 starts the declaration of the locale `S5_RS_frame`. In line 7 the name `L` is assigned to an arbitrary but fixed quadtuple of type `('a RS_frame)`. As part of the record and locale declaration the Isabelle/HOL/Isar system defines a number of functions. For example, the system introduces declarations which ensure that the expression `(r_carrier L)` is interpreted as a function call which returns the content of the first slot of the quadtuple `L` – a set of type `('a set) set`. Similarly, the expression $\mathtt{R}_\mathtt{L}$ is a function call that returns the content of the second slot of `L` – a binary relation on a set of type `('a set) set`. Finally, `(s_carrier L)` and $\mathtt{S}_\mathtt{L}$ respectively provide access to the third and fourth slot of `L`.

The record structure `L` is a computational representation of a $\mathcal{KS}$-structure with a product frame $\mathcal{K} = (\Gamma^{\mathcal{L}} \times \Sigma)$. The set `(r_carrier L)` represents the set $\Gamma^{\mathcal{L}}$ and $\mathtt{R}_\mathtt{L}$ represents the accessibility relation R^Γ

```
1   record 'a RS_frame =
2     r_carrier :: "('a set) set"
3     aR :: "'a set => 'a set => bool" (infixl "R_1" 50)
4     s_carrier :: "('a set) set"
5     aS :: "'a set => 'a set => bool" (infixl "S_1" 50)

6   locale S5_RS_frame =
7     fixes L (structure)
8     assumes
9       RCarrier: "r_carrier L ≠ ∅" and
10      R_ref [intro, simp]: "x ∈ r_carrier L ==> x R_L x" and
11      R_sym [intro]: "[|x ∈ r_carrier L; y ∈ r_carrier L; x R_L y|] ==> y R_L x" and
12      R_trans [trans]:
13        "[|x ∈ r_carrier L; y ∈ r_carrier L; z ∈ r_carrier L; x R_L y; y R_L z|] ==> x R_L z"
14    assumes
15      SCarrier: "s_carrier L ≠ ∅" and
16      S_ref [intro, simp]: "u ∈ s_carrier L ==> u S_L u" and
17      S_sym [intro]: "[| u ∈ s_carrier L; v ∈ s_carrier L; u S_L v |] ==> v S_L u" and
18      S_trans [trans]:
19        "[| u ∈ s_carrier L; v ∈ s_carrier L; s ∈ s_carrier L; u S_L v; v S_L s|] ==> u S_L s"

20  datatype 'a RS = RSC "'a set" "'a set"
21  primrec r_RS :: "'a RS => 'a set" where "r_RS (RSC r s) = r"
22  primrec s_RS :: "'a RS => 'a set" where "s_RS (RSC r s) = s"
```

Fig. 3. Declaration of product frames.

on $\Gamma^{\mathcal{L}}$; similarly, the set `(s_carrier L)` represents the set Σ and `S`$_\mathrm{L}$ represents the accessibility relation R^{Σ} on Σ. As discussed in Sec. 2.1 the product frames $\mathcal{K}$ are S5 frames. That is, $\Gamma^{\mathcal{L}}$ and Σ – represented respectively by `(r_carrier L)` and `(s_carrier L)` – are required to be non-empty and the accessibility relations R^{Γ} and R^{Σ} – represented respectively by `R`$_\mathrm{L}$ and `S`$_\mathrm{L}$ – are subject to the constraints of reflexivity, symmetry, and transitivity. These constraints are expressed as axioms of the locale `S5_RS_frame` in lines 9 – 19.

While in the somewhat declarative style of the semi-formal presentation worlds/possibilities are presented as sets of tuples of the form $\Gamma^{\mathcal{L}} \times \Sigma$, in the more operational computational representation record types are used to store tuples of sets of the form $(\Gamma^{\mathcal{L}}, \Sigma)$. The set theoretic product is generated from $(\Gamma^{\mathcal{L}}, \Sigma)$ when formulas are evaluated in specific wolds $(\gamma, \sigma) \in \Gamma^{\mathcal{L}} \times \Sigma$. In line 20 of Fig. 3 the datatype `('a RS)` is declared to hold ordered pairs of the form $(\gamma, \sigma) \in \Gamma^{\mathcal{L}} \times \Sigma$. The functions `r_RS` and `s_RS` of lines 21 and 22 provide access respectively to the first and second components of ordered pairs of type `('a RS)`.

3.2. *The propositional segment of the modal language*

Fig. 4 displays parts of the formal declaration of the propositional section of the modal language of Sec. 2.3 and Eq. 4. The code is adopted with modifications from Benzmüller (2015) and Benzmüller and Woltzenlogel Paleo (2015). Unlike Benzmüller et al. who use type declarations for worlds and domains of quantification here record structures and locales are used. This choice allows for more flexibility in the specification of the semantics of the formal language and the models of the axiomatic theory.

In the first line of this code fragment propositional formulas are declared as functions of type `('a, 'b) RS_predicate`. Functions of this type are mappings that take a product frame (represented by a variable of type `('a RS_frame)` and a particular world (represented by a variable of type `('a RS)`) to a boolean truth

```
1    type_synonym ('a, 'b) RS_predicate = "('a, 'b) RS_frame_scheme => 'a RS => bool"
2
3    abbreviation mneg:: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate" ("¬_"[52]53)
4       where "¬P ≡ (λ L w. ¬ (P L w))"
5    abbreviation mand :: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate =>
6       ('a, 'b) RS_predicate" (infixr"∧" 51)
7       where "P ∧ Q ≡ λ L w. (P L (w)) ∧ (Q L (w))"
     ...
8    abbreviation mboxR :: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate" ("□R")
9       where "□R P L w ≡ ∀γ. γ ∈ r_carrier L ∧ (r_RS w) RL γ → (P L (RSC γ (s_RS w)))"
10   abbreviation mboxR :: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate" ("□S")
11      where "□S P L w ≡ ∀σ. σ ∈ s_carrier L ∧ (s_RS w) SL σ → (P L (RSC (r_RS w) σ))"
12   abbreviation mdiaR :: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate" ("◇R")
13      where "◇R P L w ≡ ∃γ. γ ∈ r_carrier L ∧ (r_RS w) RL γ ∧ (P L (RSC γ (s_RS w)))"
14   abbreviation mdiaS :: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate" ("◇S")
15      where "◇S P L w ≡ ∃σ. σ ∈ s_carrier L ∧ (s_RS w) SL σ ∧ (P L (RSC (r_RS w) σ ))"
16   abbreviation mbox :: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate" ("□") where
17      "□P  ≡  □R(□S(P))"
```

Fig. 4. The interpretation of modal propositional formulas in `'a RS_frame` product frames. (adopted from the work by Benzmüller (2015) and Benzmüller and Woltzenlogel Paleo (2015)) (See `S5_2D_base.thy`)

value.[1] The logical operators of negation, conjunction and implication of the modal language specified in Eq. 4 are implemented in Isabelle/HOL as definitorial expressions of the form

$$\texttt{abbreviation}/\texttt{definition}\ \text{name} :: \text{type declaration}\ \texttt{where}\ \text{"defines} \equiv \text{definition"}. \tag{5}$$

In the body of the definitions, the product frames $\mathcal{K}$ of the $\mathcal{KS}$-structure represented by the record structure `L` and worlds (designated by `w`) are distributed to sub-formulas which are connected by the respective non-modal logical operators of the underlying logic HOL. As illustrations the definitions of the negation and the conjunction of the modal language are displayed in lines 3 – 7 of Fig. 4. Syntactically, the modal (world-dependent) versions of logical connectives (the defines of Eq. 5) are symbolized using bold typeface. In Isabelle/HOL expressions labeled as `abbreviation` are definitions that are automatically expanded by the system in the search for the proof of a theorem. By contrast definitorial expressions following the keyword `definition` need to be expanded explicitly in the course of a proof.

Modal operators are declared for both components of the product frames $\mathcal{K}$ of the $\mathcal{KS}$-structure represented by the record structure `L`. The modal operators $\Box^{\mathrm{R}}$ and $\Diamond^{\mathrm{R}}$ are evaluated in the standard ways with respect to the worlds in `(r_carrier L)` and the associated accessibility relation $\mathtt{R}_{\mathrm{L}}$. Similarly, the modal operators $\Box^{\mathrm{S}}$ and $\Diamond^{\mathrm{S}}$ are evaluated in the standard ways with respect to the worlds in `(s_carrier L)` and the accessibility relation $\mathtt{S}_{\mathrm{L}}$ (lines 8 – 15 of Fig. 4). Consider lines 8 and 9. It may be good practice to not only to rely on the type checking in the call of the modal operator, but to explicitly constrain the worlds in the call of the modal operator to the members of the respective carrier sets as indicated in the lines 8′ and 9′ below:

```
8′  abbreviation mboxR :: "('a, 'b) RS_predicate => ('a, 'b) RS_predicate" ("□R")
9′     where "□R P L w ≡ (r_RS w) ∈ r_carrier L ∧ (s_RS w) ∈ s_carrier L ∧
                        ∀γ. γ ∈ r_carrier L ∧ (r_RS w) RL γ → (P L (RSC γ (s_RS w)))"
```

Similarly for the other modal operators of lines 8 – 15 of Fig. 4.

[1] In the process of interpreting the declaration of a record of type `('a RS_frame)` Isabelle/HOL automatically creates a number of derived types including the type `('a, 'b) RS_frame_scheme`. The latter is the type of the actual data structure that is used to implement the record. This is the type that occurs in function declarations that are evaluated by the type system of the underlying ML language. See the work by Nipkow et al. (2002) for details.

```
1   record ('a, 'b) two_sort_RS_frame = "'a RS_frame" +
2      carrier :: "'a set"
3      e_carrier :: "'b set"
4
5   locale two_sort_S5_RS_frame = S5_RS_frame +
6   assumes
7      carrier: "carrier L ≠ ∅" and
8      carrierE: "e_carrier L ≠ ∅" and
9      Rcarrier1: "r ∈ r_carrier L ==> r ⊆ carrier L" and
10     Scarrier1: "s ∈ s_carrier L ==> r ⊆ carrier L"
```

Fig. 5. Domains of quantification. (See `S5_2D_base.thy`)

3.3. *Quantification*

As pointed out in Sec. 2.1, there are two domains of interpretations in $\mathcal{KS}$-structures: $\mathcal{D}_{ST}$ the domain of regions of spacetime and $\mathcal{D}_E$ the domain of physically possible entities. In the context of $\mathcal{KS}$-structures the domain of regions is special in the sense that the worldlines and the slicings of spacetime that constitute the carrier sets of the frame structures are regions of spacetime, i.e., members of $\mathcal{D}_{ST}$. This is encoded in the computational representation as depicted in Fig. 5.

The record type `'a RS_frame` for $\mathcal{KS}$-structures is extended by two additional slots: one slot for the set `carrier` of type `('a set)` and a second slot for the set `e_carrier` of type `('b set)` (lines 2 – 3 of Fig. 5). The type variable `'a` of the `'a RS_frame` structures of Fig. 3 is a place holder for the sort of regions while the type variable `'b` is a place holder for the sort of entities. The resulting record type is denoted `('a, 'b) two_sort_RS_frame`.

The locale `two_sort_S5_RS_frame` is declared to inherit all axioms of the locale `S5_RS_frame`. In addition four axioms are included. Two axioms ensure that both `(carrier L)` and `(e_carrier L)` are non-empty. Two additional axioms require that the sets `(r_carrier L)` and `(s_carrier L)` are both subsets of `(carrier L)`. The axioms labeled respectively: `carrier`, `carrierE`, `Rcarrier1` and `Scarrier1` (lines 5 – 10 of Fig. 5). Axioms `Rcarrier1` and `Scarrier1` illustrate the simplicity of postulating constraints among subsets of a given type. This is an important advantage of using locales and records over the built-in infrastructure for type declarations in Isabelle/HOL as it is used by Benzmüller et al.

Consider Fig. 6 which displays the declarations of the quantifiers of the modal language of Eq. 4. Again, the code is adopted with modifications from the work by Benzmüller (2015) and Benzmüller and Woltzenlogel Paleo (2015). In the figure the declaration of quantifiers that range over members and subsets of the set `carrier` are listed explicitly. The declaration of quantifiers that range over members and subsets of the set `e_carrier` are similar and are omitted here except for the declarations in lines 29 –31.

In the first line of the code fragment open formulas in which free variables can range over the types `'a`, `'a set`, `'b`, and `'b set` are declared as functions of type `('a, 'b, 'c) two_sort_RS_predicate`. Functions of this type implement mappings that take a product frame with two domains of quantification (represented by a variable of type `('a, 'b) two_sort_RS_frame` and a particular world (represented by a variable of type `('a RS)`) to a boolean truth value. The type system of Isabelle/HOL is able to infer that record variables of type `('a, 'b) RS_predicate` can be obtained from record variables of type `('a, 'b, 'c) two_sort_RS_predicate` by disregarding the two slots that are added in the declaration of `('a, 'b, 'c) two_sort_RS_predicate`. The type system thereby ensures that the declarations of negation, conjunction, disjunction, implication and logical equivalence of Fig. 4 extend to formulas of type `('a, 'b, 'c) two_sort_RS_predicate`.

Universal and existential quantifiers for variables of type `'a` and `'b` are declared as depicted in lines 3 – 14 of Fig. 6. The declaration of every quantifier has two components. One component is semantic and the other is syntactic in nature. The syntactic component of the declaration of a quantifier provides means for Isabelle/HOL to express the binding of a variable within the scope of a quantifier as a typing problem using λ-expressions (e.g., lines 6 – 8). (For details see the work by Wenzel (2017).) At the semantic level

```
1   type_synonym ('a, 'b, 'c) two_sort_RS_predicate =
2      "('a, 'b, 'c) two_sort_RS_frame_scheme => 'a RS => bool"

3   abbreviation a_mforall :: "('a => ('a, 'b, 'c) two_sort_RS_predicate) =>
4      ('a, 'b, 'c) two_sort_RS_predicate" where
5      "a_mforall P ≡ λ L w. ∀x.x ∈ carrier L → (P x) L w"
6   abbreviation a_mforallB :: "('a => ('a, 'b, 'c) two_sort_RS_predicate) =>
7      ('a, 'b, 'c) two_sort_RS_predicate" (binder "∀a"[8]9) where
8      "∀ax.P(x) ≡ a_mforall P"
    ...
9   abbreviation b_mexists :: "('b => ('a, 'b, 'c) two_sort_RS_predicate) =>
10     ('a, 'b, 'c) two_sort_RS_predicate" where
11     "b_mexists P ≡ λ L w. ∃x.x ∈ e_carrier L ∧ (P x) L w"
12  abbreviation b_mexistsB :: "('b => ('a, 'b, 'c) two_sort_RS_predicate) =>
13     ('a, 'b, 'c) two_sort_RS_predicate" (binder "∃b" [8]9) where
14     "∃bx.P(x) ≡ b_mexists P"
    ...
```

Fig. 6. The interpretation of formulas of modal predicate logic in `('a RS_frame)` product frames with type-restricted quantification (adopted from the work by Benzmüller (2015) and Benzmüller and Woltzenlogel Paleo (2015)). (See `S5_2D_base.thy`.)

(universal) quantification is restricted not only to variables of a given type but to the members of sets that are part of the structures that provide the domain of interpretation. This allows for a natural correspondence between the semi-formal definition of the semantics in Eq. 4 and the formal treatment in the computational representation. For example in line 5 the universal quantifier ($\forall_{\mathtt{a}}$) is restricted not only to variables of type `'a` but to the members of the set `(carrier L)` (of type `'a set`). The variable `L` is a place holder for a structure of type `('a, 'b) two_sort_RS_frame`. When instantiated as intended, the set `(carrier L)` will have regions of spacetime as members. Similarly for the quantifiers $\exists_{\mathtt{a}}$, $\forall_{\mathtt{b}}$, $\exists_{\mathtt{b}}$ (lines 9 – 14), $\forall_{\mathtt{A}}$, etc. in `S5_2D_base.thy`.

3.4. *Valid formulas*

In Isabelle/HOL one can then define the notion of validity. A formula `P` of type `('a, 'b) two_sort_RS_predicate` is valid in a structure `L` of type `('a, 'b) two_sort_RS_frame`, signified as $\lfloor \mathrm{P} \rfloor_{\mathtt{L}}$, if and only if `P` is true for all members of `(r_carrier L` and all members of `(s_carrier L)` (lines 1–3 of Fig. 7). On the enforced interpretation in $\mathcal{KS}$-structures the members of `(r_carrier L)` represent dynamically possible worldlines in $\Gamma^{\mathcal{L}}$ and the members of `(s_carrier L)` represent kinematically possible slicings Σ of spacetime. A common textbook notation that is equivalent to $\lfloor \mathrm{P} \rfloor_{\mathtt{L}}$ is ($\models_{\mathtt{L}}$ P). Since `L` is arbitrary but fixed one also writes ($\lfloor \mathrm{P} \rfloor$ and $\models$ P).

Within this framework Isabelle/HOL is able to prove the usual theorems about the logical interrelations between $\Box$ and $\diamond$ (e.g., line 4 of Fig. 7), the validity of the S5 axioms (e.g., T in lines 6), as well as the Barcan formula (line 7). In addition one can prove that the order of the quantifiers $\Box^{\mathrm{R}}$ and $\Box^{\mathrm{S}}$ is immaterial (line 5). In contrast to the semi-formal presentation of the theory, in the computational representation all of these properties have to be proved for all types of variables and the associated typed quantifiers $\forall_{\mathtt{a}}, \forall_{\mathtt{b}}, \forall_{\mathtt{A}}, \exists_{\mathtt{a}}, \ldots$ as well as for all types of modal operators including $\Box^{\mathrm{R}}, \Box^{\mathrm{S}}, \Box, \diamond^{\mathrm{R}}, \ldots$. For example, line 4 displays a lemma for $\forall_{\mathtt{a}}$ and $\Box^{\mathrm{R}}$, while line 5 displays a lemma for $\forall_{\mathtt{b}}$, $\Box^{\mathrm{R}}$ and $\Box^{\mathrm{S}}$. All the properties of modal formulas with unary predicates immediately generalize to formulas with n-ary predicates as shown for the Barcan formula in line 7. The term `(P x)` in line 6 is of type `('a, 'b) two_sort_RS_predicate` and so is the term `(P x y)` in line 7. Of course, the term `P` in line 6 is of type `'a => ('a, 'b) two_sort_RS_predicate` while the term `P` in line 7 is of type `'a => 'a => ('a, 'b) two_sort_RS_predicate`.

```
abbreviation mvalid :: "('a, 'b, 'c) two_sort_RS_frame_scheme =>
    ('a, 'b, 'c) two_sort_RS_predicate => bool" ("⌊_⌋_1"[7]8) where
    "⌊ P ⌋_L ≡ ∀γ. ∀σ. γ ∈ r_carrier L ∧ σ ∈ s_carrier L → (P L (RSC γ σ))"

lemma (in two_sort_S5_RS_frame) "⌊∀_a x. (□^R(Px))↔(¬◇^R(¬Px))⌋" by blast
lemma (in two_sort_S5_RS_frame) "⌊∀_b x. (□^S(□^R(Px)))↔(□^R(□^S(Px)))⌋" by force

lemma (in two_sort_S5_RS_frame) "⌊∀_a x. □(Px)→(Px)⌋" using R_ref S_ref by auto
lemma (in two_sort_S5_RS_frame) "⌊(∀_a x.∀_b y. □(Pxy))↔(□(∀_a x.∀_b y. (Pxy)))⌋" by fast
```

Fig. 7. Validity of formulas of type `('a, 'b) two_sort_RS_predicate` in structures of type `('a, 'b) two_sort_RS_frame`. (adopted from the work by Benzmüller (Benzmüller, 2015; Benzmüller and Woltzenlogel Paleo, 2015)) (See `S5_2D_base.thy`)

Theorems and lemmata in Isabelle/HOL/Isar are to be read as follows: the keywords `lemma`/`theorem` are synonymous; the statement enclosed by the quotation marks is the actual statement that is proved; in parenthesis in front of the statement is the locale in the context of which (type declarations, definitions, and axioms associated with the locale) the statement was proved; following the statement enclosed by the keywords `using` and `by` are names of axioms and theorems that are used in the proof; and finally following the keyword `by` the (built in) proof method (`auto`, `fast`, etc.) that was used in the proof is listed. If no axioms and definitions are listed in the `using` section of a lemma/theorem then the proof was found using axioms and theorems in Isabelle/HOL, simplification rules of the lambda calculus, and by expanding definitonal expressions labeled as abbreviations.

This concludes the setup of the formal language for the axiomatic theory. The computational realization is the formalized theory in `S5_2D_base.thy`. In what follows the ontology will be developed hierarchically. At each level of the theory hierarchy three formal components are introduced in a way that mirrors the methodology in this section: (1) at every level new axiomatic primitives are introduced by extending the record structures that represent the $\mathcal{KS}$-structures; (2) axioms characterizing the new primitive are collected in corresponding locales; (3) at the level of the modal language of the formal ontology every formula is typed. The type of a formula is determined by the class of structures in which it is interpreted and correspondingly, the set of axioms that are associated with these structures. The hierarchical development of the ontology and the correspondence of the three components of the theory at every level are summarized in Table 2 on page 31.

The automatically generated presentation of the fully formalized and computationally verified formal ontology can be found at: `http://www.buffalo.edu/~bittner3/Theories/OntologyCM/`.

4. A mereology of space-time and its computational realization

There are a number of formalizations of mereological theories (Simons, 1987; Varzi, 2003; Tarski and Givant, 1999; Champollion and Krifka, 2015). The logical relations between the various systems are mostly understood. What is relevant to this paper is that the aim of a computational realization may affect the choice of a given system within a 'space' of logically equivalent systems.

For example, Bittner (2018) employs a system of mereology that emphasizes an algebraic (lattice-theoretic) view of mereology over a more standard (order-theoretic) view (Champollion and Krifka, 2015). The algebraic view of mereology is attractive in the context of this paper because the formal environment of Isabelle/HOL already provides a fully formalized computational realization of lattice theory in `HOL/Algebra/Lattice.thy` (Ballarin, 2017). Within the context of an algebraic view of mereology then a system by Krifka (1998) was selected in the semi-formal presentation of Bittner (2018). This system is based on the primitives of mereological union and mereological intersection. It facilitates a simple and compact presentation of mereological notions in a first order language in the context of a paper where mereology is not the focus of the attention.

The semi-formal presentation of the mereology used by Bittner (2018) is briefly reviewed in subsection 4.1.1. The computational realization of the mereology then has of three components: (1) The mereological axioms are expressed in a lattice-theoretic framework using a non-modal second order language and explicit reference to $\mathcal{KS}$-structures by means of the record structures and associated locales of Isabelle/HOL/Isar; (2) A formalized proof is provided in which it is verified that these axioms are satisfied in the computational realization of the example model; (3) The mereological axioms are lifted to the (mostly) first order modal level of the formal presentation. At this level the axioms and definitions of the semi-formal presentation of subsection 4.1.1 are recovered as theorems.

4.1. Mereology in a lattice-theoretic framework

On the algebraic view of mereology is natural to reuse the fully formalized computational realization of lattice theory in `HOL/Algebra/Lattice.thy` (Ballarin, 2017). In the highly expressive framework of HOL it is also natural to declare lattice structures in a way that mirrors the ways in which lattices are introduced in mathematics. Unlike lattices in `HOL/Algebra/Lattice.thy`, however, mereological structures lack a minimal element. For this reason a modified version of `HOL/Algebra/Lattice.thy` is used here.

4.1.1. The semi-formal presentation of mereology

To capture the mereological structure of spacetime regions in the semi-formal presentation of Bittner (2018) the primitive binary operation $\sqcup : \mathcal{D}_{ST} \times \mathcal{D}_{ST} \to \mathcal{D}_{ST}$ is introduced in the (first-order) object language of the formal theory. On the intended interpretation in $\mathcal{KS}$-structures $\sqcup$ is the mereological union of regions u_1 and u_2. More precisely, $\sqcup$ is interpreted as an operation that yields the least upper bound $\bigsqcup : \mathcal{P}(\mathcal{D}_{ST}) \to \mathcal{D}_{ST}$ of the set $\{\mathsf{u}_1, \mathsf{u}_2\}$ with respect to the ordering imposed on $\mathcal{D}_{ST}$ by $\sqsubseteq$ (Champollion and Krifka, 2015). The second primitive of the semi-formal presentation of the formal theory of Bittner (2018) is the ternary functional relation $\sqcap$. On the intended interpretation in $\mathcal{KS}$-structures $\sqcap$ is the mereological intersection that holds between regions u_1, u_2 and u_3 if and only if the greatest lower bound $\bigsqcap : \mathcal{P}(\mathcal{D}_{ST}) \to (\mathcal{D}_{ST} \cup \emptyset)$ of the set $\{\mathsf{u}_1, \mathsf{u}_2\}$ exists and $\mathsf{u}_3 = \bigsqcap\{\mathsf{u}_1, \mathsf{u}_2\}$ (Eq. 6) (Champollion and Krifka, 2015).[2]

$$
\begin{array}{llll}
\mathsf{V}(\sqcup) & = \underline{\sqcup} =_{df} & \{\langle \mathsf{u}_1, \mathsf{u}_2, \mathsf{u}_3, \kappa\rangle \in \mathcal{D}_{ST} \times \mathcal{D}_{ST} \times \mathcal{D}_{ST} \times \mathcal{K} \mid \mathsf{u}_3 = \bigsqcup\{\mathsf{u}_1, \mathsf{u}_2\}\} \\
\mathsf{V}(\sqcap) & = \overline{\sqcap} =_{df} & \{\langle \mathsf{u}_1, \mathsf{u}_2, \mathsf{u}_3, \kappa\rangle \in \mathcal{D}_{ST} \times \mathcal{D}_{ST} \times \mathcal{D}_{ST} \times \mathcal{K} \mid \mathsf{u}_3 = \bigsqcap\{\mathsf{u}_1, \mathsf{u}_2\}\}
\end{array} \tag{6}
$$

The binary predicate of parthood, $P\ uv$, is defined to hold if and only if the union of u and v is identical to v (D_P). The predicate *ST* holds of a region which has all regions as parts (D_{ST}). Proper parthood (*PP*), overlap (O), and summation are defined in the standard ways in D_O and D_{Sum} (Simons, 1987).

$$
\begin{array}{ll}
D_P\ \ P\ uv \equiv u \sqcup v = v & \\
D_{PP}\ PP\ uv \equiv P\ uv \land v \neq u & D_{ST}\ \ ST\ u \equiv (v)P\ vu \\
D_O\ \ O\ uv \equiv (\exists w)(P\ wu \land P\ wv) & D_{Sum}\ Sum\ xA \equiv (\forall w)(O\ xw \leftrightarrow (\exists z)(z \in A \land O\ zw))
\end{array}
$$

On the enforced interpretation in $\mathcal{KS}$-structures (Eq. 1) the parthood predicate P holds of the relation $\sqsubseteq$ on $\mathcal{D}_{ST}$; the predicate *ST* holds of the maximal element $\mathcal{ST}$ of $\mathcal{D}_{ST}$; the overlap predicate is true if two regions share a member of $\mathcal{D}_{ST}$; and the *Sum* predicate holds of least upper bounds of some non-empty subsets of $\mathcal{D}_{ST}$ as indicated in Eq. (7).

$$
\begin{array}{lll}
\mathsf{V}(P) & = & \{\langle \mathsf{u}_1, \mathsf{u}_2, \kappa\rangle \in \mathcal{D}_{ST} \times \mathcal{D}_{ST} \times \mathcal{K} \mid \mathsf{u}_1 \sqsubseteq \mathsf{u}_2\} \\
\mathsf{V}(ST) & = & \{\langle \mathcal{ST}, \kappa\rangle \in \mathcal{D}_{ST} \times \mathcal{K} \mid \mathcal{ST} = \bigsqcup \mathcal{D}_{ST}\} \\
\mathsf{V}(O) & = & \{\langle \mathsf{u}, \mathsf{v}, \kappa\rangle \in \mathcal{D}_{ST} \times \mathcal{D}_{ST} \times \mathcal{K} \mid \exists \mathsf{w} \in \mathcal{D}_{ST} : \mathsf{w} \sqsubseteq \mathsf{u} \land \mathsf{w} \sqsubseteq \mathsf{v}\} \\
\mathsf{V}(Sum) & \subseteq & \{\langle \mathsf{u}, \mathsf{A}, \kappa\rangle \in \mathcal{D}_{ST} \times \mathcal{P}(\mathcal{D}_{ST}) \times \mathcal{K} \mid \mathsf{A} \neq \emptyset \land \mathsf{u} = \bigsqcup \mathsf{A}\}
\end{array} \tag{7}
$$

Axioms are introduced requiring that $\sqcup$ is idempotent, associative, commutative (A1 – A3) and that there exists a spacetime region which has all regions as parts (A4). Furthermore an axiom of separation (Cham-

[2]This presentation traces over the difficulties that arise in the context of a mereology of regions which are represented by manifolds rather than by topologically regular sets. See also A.1 and (Butterfield, 2007; Arnold, 1997).

pollion and Krifka, 2015) is required to hold (A5). Introducing $\sqcap$ as a relational primitive implicitly acknowledges that mereological intersections do not always exist. An axiom is introduced requiring that an intersection of two overlapping regions always exists (A6).

$$
\begin{array}{ll}
A1\ u \sqcup u = u & A5\ \mathit{PP}\ uv \rightarrow (\exists w)(\neg O\ uw \wedge v = u \sqcup w) \\
A2\ u \sqcup (v \sqcup w) = (u \sqcup v) \sqcup w & A6\ O\ uv \rightarrow (\exists w)(\sqcap\ uvw) \\
A3\ u \sqcup v = v \sqcup u & A7\ u \sqcup v = w \rightarrow \Box(u \sqcup v = w) \\
A4\ (\exists u)(\mathit{ST}\ u) & A8\ \sqcap\ uvw \rightarrow \Box(\sqcap\ uvw)
\end{array}
$$

As specified in Eq. 6 the mereological unions and intersections are the same at all possible worlds. This explicates at the level of the interpretation of the formal theory that the mereological structure of spacetime is *absolute* in the sense that it is the same on all physical possibilities and slicings. In the object language this is mirrored in axioms A7 and A8.

4.1.2. Computational realization in Isabelle/HOL/Isar

The computational realization of the mereology of the previous section in Isabelle/HOL/Isar can be found in the file `Plattice.thy` (which is adopted from `HOL/Algebra/Lattice.thy`). Axiomatic primitives of the computational realization are introduced in two steps: (i) by extending record structures of type `('a, 'b) two_sort_RS_frame` and (ii) by extending the locale `two_sort_S5_RS_frame` of Fig. 5. A record type R_2 is an extension of record type R_1 if R_2 includes all the 'slots' of R_1 and R_2 has at least one 'slot' that is not included in R_1. Similarly, a locale L_2 extends the locale L_1 if L_2 includes all the axioms of L_1 and adds further axioms. This is summarized for all the primitives and axioms of the formalized ontology in Table 2 on page 31.

Following the standard lattice theory in `HOL/Lattice.thy` there is a single primitive predicate for partial orderings. The declarations that introduce this primitive in the computational realization are depicted in Fig. 8. First the record type `('a, 'b) porder_two_sort_RS_frame` is declared that includes a slot for the function `le :: "'a => 'a => bool"` as depicted in line 2 of the figure. To emphasize that `le` is the computational representation of parthood relation in $\mathcal{KS}$-structures the symbol $\sqsubseteq$ is used to refer to `le` in the declarations that follow. The locale `S5_RS_2S_partial_order` states the axioms of reflexivity, antisymmetry, and transitivity for $\sqsubseteq$ and mirrors the declarations of `HOL/Algebra/Lattice.thy`.

```
1   record ('a, 'b) porder_two_sort_RS_frame = "('a, 'b) two_sort_RS_frame" +
2     le :: "'a => 'a => bool" (infixl "⊑_1" 50)

3   locale S5_RS_2S_partial_order = two_sort_S5_RS_frame L for L (structure) +
4   assumes
5    le_refl [intro, simp]: "x ∈ carrier L ==> x ⊑ x" and
6    le_antisym [intro]:
7      "[| x ⊑ y; y ⊑ x; x ∈ carrier L; y ∈ carrier L |] ==> x = y" and
8    le_trans [trans]:
9      "[| x ⊑ y; y ⊑ z; x ∈ carrier L; y ∈ carrier L; z ∈ carrier L |] ==> x ⊑ z"

10  definition lless :: "[_, 'a, 'a] => bool" (infixl "⊏_1" 50) where
11     "x ⊏_L y ≡ x ⊑_L y ∧ x ≠ y"
12  definition overlap :: "[_, 'a, 'a] => bool" (infixl ".O_1" 70) where
13     "x .O_L y ≡ (∃ z. z ∈ carrier L ∧ z ⊑_L x ∧ z ⊑_L y)"
    ...
```

Fig. 8. Partial orderings. Adopted from the work by Ballarin (2017). (See `Plattice.thy`)

In contrast to the semi-formal presentation of the theory in Sec. 4.1.1, in the higher-order language of the computational representation the predicate $\sqcup$ is defined rather than introduced as a primitive. The definition of $\sqcup$ in line 7 of Fig. 9 expresses the usual lattice theoretic understanding of this operation in terms of least upper bounds. The necessary declarations are displayed in lines 1 – 6. These definitions explicitly formalize what is expressed semi-formally in the specification of the intended interpretation $\mathsf{V}(\sqcup)$ in Eq. 6. In writing the axioms A1 – A3 for $\sqcup$ as equations one implicitly assumes that binary mereological unions always exist. In the computational representation it is explicitly postulated that least upper bounds of arbitrary non-empty subsets (of `(carrier L)`, i.e., $\mathcal{D}_{ST}$) always exist (lines 9 – 10 of Fig. 9). In the mereological context this amounts to postulating the existence of mereological unions for arbitrary non-empty subsets of $\mathcal{D}_{ST}$.

```
1   definition Upper :: "[_, 'a set] => 'a set" where
2      "Upper L A = {u. (ALL x. x ∈ A ∩ carrier L -> x ⊑L u)} ∩ carrier L"
3   definition least :: "[_, 'a, 'a set] => bool" where
4      "least L l A ≡ A ⊆ carrier L ∧ l ∈ A ∧ (ALL x : A. l ⊑L x)"
5   definition sup :: "[_, 'a set] => 'a" ("⨆1_" [90] 90) where
6      "⨆LA = (SOME x. least L x (Upper L A))"
7   definition join :: "[_, 'a, 'a] => 'a" (infixl "⊔1" 65) where "x ⊔L y = ⨆L{x, y}"
8
9   locale complete_upper_semilattice = S5_RS_2S_partial_order L for L (structure) +
10  assumes sup_exists:
11     "[| A ≠ ∅; A ⊆ carrier L |] ==> EX s. least L s (Upper L A)"
```

Fig. 9. Formalized axioms for complete upper semi-lattices. Adopted from the work by Ballarin (2017) (See `Plattice.thy`)

While in the semi-formal presentation in Sec. 4.1.1 the predicate $\sqcap$ is a primitive, the corresponding predicate `is_meet` of the HOL-based computational representation is introduced by definition. The definition of `is_meet` in lines 3 and 4 of Fig. 10 mirrors the usual lattice-theoretic definitions in terms of greatest lower bounds (see definition of `greatest` and `Lower` in `Plattice.thy`). As in the case of the operation $\sqcup$ above these declarations explicitly formalize what is expressed semi-formally in the specification of the intended interpretation $\mathsf{V}(\sqcap)$ in Eq. 6. The absence of a minimal element in mereology means that not every set of domain members has a greatest lower bound. To single out pairs of domain members that (when jointly considered as a set of cardinality two) do have a greatest lower bound the relation of overlap is used in the usual ways as specified in the declaration of the locale `partial_lower_semilattice` in lines 7 – 8 of Fig. 10. This corresponds to A6 in the semi-formal presentation.

```
1  definition is_inf :: "[_, 'a set,'a] => bool" where
2     "is_inf L A a ≡ greatest L a (Lower L A)"
3  definition is_meet :: "[_, 'a, 'a, 'a] => bool" where
4     "is_meet L x y z ≡ is_inf L {x, y} z"

5  locale partial_lower_semilattice = S5_RS_2S_partial_order L for L (structure) +
6  assumes inf_of_two_exists:
7     "[| x ∈ carrier L; y ∈ carrier L; x .OL y |] ==>
8        EX s. greatest L s (Lower L {x, y})"
```

Fig. 10. Formalized axioms for partial lower semi-lattice. Adopted from the work by Ballarin (2017). (See `Plattice.thy`)

The axioms that form a general extensional mereology (Simons, 1987) are collected in the locale `S5_RS_2S_GEM` (Fig. 11) which inherits all the axioms and definitions of the locales `complete_upper_semilattice` and `partial_lower_semilattice` via the locale `partial_lattice` and adds two further axioms. The axiom `noBot` ensures that the set `carrier L` (i.e., $\mathcal{D}_{ST}$) does not have a minimal element. This requirement is implicit in the semi-formal presentation. Corresponding to axiom A5 the remainder principle `RP` is included.

```
locale partial_lattice = complete_upper_semilattice L +
   partial_lower_semilattice L for L (structure)

locale S5_RS_2S_GEM = partial_lattice L for L (structure) +
assumes noBot: "(greatest L l (Lower L (carrier L))) ==> l ∉ carrier L" and
   RP: "[| x ∈ carrier L; y ∈ carrier L; x ⊏ y|] ==>
       (∃z ∈ (carrier L). ¬(z .O x) ∧ (z ⊔ x = y))"
```

Fig. 11. Formalized axioms for the mereology of spacetime. (See `Plattice.thy`)

4.2. *Validity in the example model*

Isabelle/HOL provides computational tools for generating models for sets of formulas (e.g., `nitpick` (Blanchette, 2017)). For the development of an axiomatic theory this is helpful to avoid inconsistent sets of axioms, i.e., sets of axioms that do not have models at all. From the point of formal ontology development this is not sufficient. The whole point of developing a formal ontology is it to create axiomatic systems that constrain sets of models to a subset of *intended* models. Unfortunately, this is rather difficult and without formal languages with significant expressive power often impossible.

However, there is a 'space' of intermediate possibilities between the two extrema merely of demonstrating the consistency of an axiomatic theory on one hand and demonstrating that the theory exactly constrains a certain class of models on the other hand. This intermediate 'space' is of interest from the perspective of developing formal ontologies. Often, when developing a formal ontology one has a set of prototypical examples in mind that guide the development of the formal theory. For example, in the specification document of Basic Formal Ontology (Smith, 2016) the authors provide prototypical examples as parts of the Elucidations that complement the semi-formal definitions and axioms. When considered as formal models these examples must satisfy all the axioms of the corresponding formal ontology. In the context of the formal ontology considered in this paper the example of Sec. 2.2 plays this role.

In the remainder of this subsection first the formal specification of the example of Sec. 2.2 will be illustrated and then the proof that demonstrates that this model satisfies the mereological axioms is discussed.

4.2.1. *The computational realization of the example model (Mereology)*

There is a difficulty when specifying a formal model for verifying the consistency of a set of axioms: the model itself cannot be specified means of axioms. This is because the axioms specifying the model may be inconsistent or they may fail to sufficiently specify that model and thus fail to serve the purpose of verifying the original set of axioms. As a solution to this problem Isabelle/HOL offers a *definitional* approach to the specification of formal models (Nipkow et al., 2002). Within this definitional approach one assumes that the axiomatization of HOL in Isabelle/HOL is consistent and can be extended by means of a restricted class of definitions and declarations in ways that cannot lead to inconsistencies. In the formal specification of the example model of Sec. 2.2 only the kinds of definitions and declarations are used that fall within the definitional approach.

Consider Fig. 12. Lines 1 – 10 depict the declarations that constitute the computational representation of the mereology associated with the spacetime consisting of six atomic regions as depicted in the left of Fig. 2. Lines 10 – 20 depict the declarations that represent the slicings of spacetime in the middle of Fig. 2. The accessibility relation among slicings of spacetime is declared in lines 21–22. Declarations of the dynamically possible worldlines in conjunction with the declaration of the associated accessibility relation are depicted in lines 23 – 28. In lines 29–32 a record structure `ST_frame` of type `"(Reg, Reg) porder_two_sort_RS_frame"` is declared which slots are 'filled' with the respective sets and relations of the example model as declared in the figure.

In addition one can verify intuitions about the enforced model by proving simple lemmata. Many examples can be found in the file `ST_model_base.thy`. Of course many of those lemmata are trivial in the context of the simple example model but they illustrate how easy it is to prove properties of the set-theoretic structures that serve as models of the axiomatic theory.

```
datatype Xcoord = ZeroX | OneX
datatype Tcoord = ZeroT | OneT | TwoT
datatype CoordT = CoordC Xcoord Tcoord
abbreviation c_00 :: "CoordT" where "c_00 ≡ CoordC ZeroX ZeroT"
...
abbreviation c_12 :: "CoordT" where "c_12 ≡ CoordC OneX TwoT"
type_synonym Reg = "CoordT set"
abbreviation top_of_m_set :: "Reg" where
   "top_of_m_set ≡ {c_00,c_01,c_02,c_10,c_11,c_12}"
abbreviation m_set :: "Reg set" where "m_set ≡ { x. x ⊆ top_of_m_set ∧ x ≠ ∅ }"

abbreviation ts0 :: "Reg" where "ts0 ≡ {c_00,c_10}"
abbreviation ts1 :: "Reg" where "ts1 ≡ {c_01,c_11}"
abbreviation ts2 :: "Reg" where "ts02≡ {c_02,c_12}"
abbreviation ts0_M :: "Reg" where "ts0_M ≡ {c_10}"
abbreviation ts1_M :: "Reg" where "ts1_M ≡ {c_00,c_11}"
abbreviation ts2_M :: "Reg" where "ts02_M ≡ {c_01,c_12}"
abbreviation ts3_M :: "Reg" where "ts03_M ≡ {c_22}"
abbreviation ts_set_M_0 :: "Reg set" where "ts_set_M_0 ≡{ts0,ts1,ts2}"
abbreviation ts_set_M_1 :: "Reg set" where "ts_set_M_1 ≡{ts0_M,ts1_M,ts2_M,ts3_M}"
abbreviation ts_set_M :: "(Reg set) set" where "ts_set_M ≡{ts_set_M_0,ts_set_M_1}"
abbreviation ar_TS_M:: "Reg set => Reg set => bool" where
   "ar_TS_M ≡λ r s. r ∈ ts_set_M ∧ s ∈ ts_set_M"

abbreviation wlA_0 :: "Reg" where "wlA_0 ≡{c_00,c_01,c_02}"
abbreviation wlA_1 :: "Reg" where "wlA_1 ≡{c_10,c_11,c_12}"
abbreviation wlCpmpl_0 :: "Reg set" where "wlCompl_0 ≡{wlA_0,wlA_1}"
abbreviation wl_Phys_Possible :: "(Reg set) set" where "wl_Phys_Possible ≡{wlCompl_0}"
abbreviation ar_WL:: "Reg set => Reg set => bool" where
   "ar_WL ≡λ r s. r ∈ wl_Phys_Possible ∧ s ∈ wl_Phys_Possible"

abbreviation ST_frame :: "(Reg, Reg) porder_two_sort_RS_frame" where
   "ST_frame ≡(|r_carrier = wl_Phys_Possible, aR = ar_WL,
              s_carrier = ts_set_M, aS = ar_TS_M,
              carrier = m_set, e_carrier = m_set, le = op ⊆ |)"
```

Fig. 12. Space and regions of spacetime, timeslices, and dynamically possible worldlines of the example model. (`ST_model_base.thy`)

4.2.2. *Using records and locales for linking axioms and models*

The structure `ST_frame` is an *enforced* model of the computational realization of the spacetime mereology which axioms are collected in the locales `S5_RS_2S_GEM`, `partial_lattice`, `complete_upper_semi-lattice`, `partial_lower_semilattice`, `S5_RS_2S_partial_order`, `two_sort_S5_RS_frame`, and `S5_RS_frame`. The top level of the formalized proof that shows that `ST_frame` (abbreviated in the proof as `?L`) indeed has all the required properties is displayed in Fig. 13. When processing the declaration of the locale `S5_RS_2S_GEM` the underlying proof assistant generates a number of proof obligations in form of the rule `S5_RS_2S_GEM.intro` that, when fulfilled for a given structure (`ST_frame` in this case) constitute a proof that the structure at hand satisfies all the axioms that are associated with the respective locale (Lines 5 – 15 of Fig. 13). The layout of the proof in addition illustrates that the proof assistant also ensures that all the axioms of the parent locales are satisfied: line 3 shows that all the axioms associated with the locale `partial_lattice` are satisfied as a consequence of the theorem `m_set_is_partial_lattice_M` and the rules of inference that are available to theorem prover `auto`. The complete proof can be found in the file

ST_model_proof_MereologyOnly.thy.

```
theorem (in S5_RS_2S_GEM) "S5_RS_2S_GEM (ST_frame)" (is "S5_RS_2S_GEM ?L")
proof (rule S5_RS_2S_GEM.intro)
  show "partial_lattice ?L" using m_set_is_partial_lattice_M by auto
next
  show "S5_RS_2S_GEM_axioms ?L"
  proof
    show "carrier ?L ≠ ∅" by auto
  next
    show "⋀l. greatest ?L l (Lower ?L (carrier ?L)) ==> l ∉ carrier ?L"
      using greatest_lower_not_in_carrier_M by blast
  next
    show "⋀x y. x ∈ carrier ?L ==> y ∈ carrier ?L ==> x ⊏?L y ==>
      ∃z∈carrier ?L. ¬ z .O?L x ∧ z ⨆?L x = y"
      using remainder_principle_M by blast
  qed
qed
```

Fig. 13. The record structure `ST_frame` satisfies the axioms of the locale `S5_RS_2S_GEM`. $\bigwedge$ is an alternative way of expressing universal quantification. (See `ST_model_proof_MereologyOnly.thy`)

4.3. Lifting to the modal level

In order to present the formalized theory in the more concise and conceptually clearer form that is employed in the semi-formal presentation of Sec. 4.1.1, the rather complex and hard to read axioms, definitions, and theorems from the level of axiomatization are lifted to the modal language of Sec. 3. In the modal language all explicit references to semantic features such as the record structures that provide the enforced interpretation are hidden. This information, although transparent to the user, remains available to the system (and can be used in the proofs). Explicit typing of formulas (discussed below) in conjunction with formal features such as currying and the rules of α-, β-, and γ- reduction of the underlying lambda calculus (part of the rules of inference in HOL) allow the system to route implicit arguments through the proofs in a way that is transparent to the user (for details see the work by Benzmüller (2015) and Benzmüller and Woltzenlogel Paleo (2015)). An illustration of how the modal formula `"⌊□(∀a x. P_M x x)⌋` stated in the context of the locale `S5_RS_2S_GEM` is 'understood' by the system, is displayed in Fig. 14.

```
1  theorem (in S5_RS_2S_GEM) "⌊□(∀a x. P_M x x)⌋"

2  theorem "∀γ σ. γ ∈ r_carrier L ∧ σ ∈ s_carrier L →
3      (∀γ'. γ' ∈ r_carrier L ∧ r_RS (RSC γ σ) R γ' →
4          (∀σ'. σ' ∈ s_carrier L ∧ s_RS (RSC γ' (s_RS (RSC γ σ))) S σ' →
5              (∀x.x ∈ carrier L →
6                  P_M x x L (RSC (r_RS (RSC γ' (s_RS (RSC γ σ)))) σ'))))"
```

Fig. 14. Lines 2 – 6 illustrate how Isabelle/HOL expands the theorem stated in line 1.

As an illustration of the typing of formulas consider lines 1 and 2 of Fig. 15. Every atomic formula in the lifted mereology is a function of type `('a, 'b, 'c) M_porder_predicate` which has two arguments: (1) a record structure of type `('a, 'b, 'c) porder_two_sort_RS_frame` which holds the domain of interpretation including the domains of the variables, the interpretation of the axiomatic primitives, possible worlds, was well as accessibility relations; and (2) the current world of type `('a RS)`. Like every closed atomic formula in a two-valued logic expressions of type `('a, 'b, 'c) M_porder_predicate` are functions that evaluate to the boolean values of true or false.

For example, the predicate `P_M` is a function of type `'a => 'a => ('a, 'b, 'c) M_porder_predicate` (line 3 of Fig. 15). Therefore, the expression `(P_M x y)` is a function that, when presented with an argument of type `('a, 'b, 'c) porder_two_sort_RS_frame` and an argument of type `('a RS)` yields a boolean truth value, the computation of which is specified in line 4 – 5. The body of the definition of every predicate links this predicate to its *enforced* interpretation. In line 4 the predicate `P_M` is linked to the predicate $\sqsubseteq_{\mathrm{L}}$. While the typing system ensures that all arguments are of the correct type, every definition ensures that every argument is a member of the correct carrier set: in lines 4 – 5 it is verified that `x,y ∈ carrier L`. Complex expressions such as $\forall_{\mathrm{a}}$`x. P_M x x`, $\Box(\forall_{\mathrm{a}}$`x. P_M x x)` are functions of type `('a, 'b, 'c) M_porder_predicate => ('a, 'b, 'c) M_porder_predicate` and are evaluated as specified in the computational realization of the formal language in Sec. 3. Similarly for the predicate `J_M` (lines 6 – 8 of Fig. 15) and all the other predicates in the semi-formal presentation of the theory.

Using the definitions of the predicates of the modal language one can prove: (i) all the axioms stated in the semi-formal presentation are theorems in the formalized modal presentation. Those theorems state that the respective formulas are valid the class of underlying structures in the sense defined in lines 1 – 3 of Fig. 7. For example, the lemmata in lines 11 and 12 of Fig. 15 are formalized versions of axioms A1 and A7 of Sec. 4.1.1. (ii) One can prove formalized versions of the theorems listed in the semi-formal development of the formal theory. For example, the lemmata inlines 15 and 16 of the figure are formalized versions of theorems T1 and T3 of Bittner (2018). (iii) Finally, one has to prove that all the definitions in the semi-formal presentation are provable as logical equivalences in the formalized theory. Lines 13 and 14 of Fig. 15 prove definition D_P as a logical equivalence.

As specified in the definition of `J_M`, mereological unions are the same at all possible worlds. This is reflected by the theorem in line 12 of Fig. 15. Similar patterns hold for all purely mereological predicates in `PLattice_lifted_theory.thy`.

```
1    type_synonym ('a, 'b, 'c) M_porder_predicate =
2      "('a, 'b, 'c) porder_two_sort_RS_frame_scheme => 'a RS => bool"

3    definition P_M :: "'a => 'a => ('a, 'b, 'c) M_porder_predicate" where
4      "P_M x y L w ≡ x ⊑L y ∧ x ∈ carrier L ∧ y ∈ carrier L ∧
5                   (r_RS w ∈ r_carrier L) ∧ (s_RS w ∈ s_carrier L)"
6    definition J_M :: "'a => 'a => ('a, 'b, 'c) M_porder_predicate" where
7      "J_M x y z L w ≡ z = x ⊔L y ∧ x ∈ carrier L ∧ y ∈ carrier L ∧ z ∈ carrier L ∧
8                   (r_RS w ∈ r_carrier L) ∧ (s_RS w ∈ s_carrier L)"
     ...
9    context S5_RS_2S_GEM
10   begin
11     lemma "⌊□(∀ax. J_M x x x)⌋" unfolding J_M_def using join_idemp by simp
12     lemma "⌊□(∀ax y z. J_M x y z → (□(J_M x y z)))⌋" unfolding J_M_def by simp
     ...
13     lemma "⌊□(∀ax1 x2. (P_M x1 x2) ↔ (J_M x1 x2 x2))⌋" unfolding P_M_def J_M_def
14       using le_iff_join by auto
     ...
15     lemma "⌊□(∀ax. P_M x x)⌋" unfolding P_M_def using le_refl by simp
16     lemma "⌊□(∀ax y z. P_M x y ∧ P_M y z → P_M x z)⌋" unfolding P_M_def
17       using le_trans by auto
18   end
```

Fig. 15. The excerpts from the lifted mereology of `PLattice_lifted_theory.thy`

5. Timeslice mereology

Mirroring the methodology of Sec. 4 the formalization of the primitive of a time slice in the context of the mereology developed above is discussed. As above the computational realization of the timeslice mereology of Bittner (2018) is presented in four steps: (1) the semi-formal presentation of the theory is reviewed; (2) the axioms are expressed in a non-modal second order language with explicit reference to $\mathcal{KS}$-structures using the record structures and associated locales of Isabelle/HOL/Isar; (3) A proof is provided that these axioms are satisfied in the computational realization of the example model; (4) The axioms and definitions of the second order language are lifted to the first order modal level of the formal presentation.

5.1. Semi-formal presentation

In the semi-formal presentation of the theory by Bittner (2018) a third primitive is added to the axiomatic theory: the unary predicate *TS*. On the enforced interpretation in $\mathcal{KS}$-structures (Eq. 1), *TS* holds of time slices $\sigma_{\mathsf{t}}(\mathcal{T})$ induced by the $\mathcal{T}$-slicing σ:

$$\mathsf{V}(TS) \quad = \quad \mathsf{TS} =_{df} \{\langle \mathsf{u}, \langle \gamma, \sigma \rangle\rangle \in \mathcal{D}_{ST} \times \mathcal{K} \mid \exists \mathsf{t} \in \Re : \mathsf{u} = \sigma_{\mathsf{t}}(\mathcal{T})\} \tag{8}$$

In terms of the primitive time slice predicate spatial and soatio-temporal regions are defined: Spatial regions are regions that are parts of some time slice (D_{SR}). Spatio-temporal regions are regions that overlap two distinct time slices (D_{STR}). Two regions are *simultaneous* if and only if they are parts of the same time-slice (D_{SIMU}).

$$\begin{array}{ll} D_{SR} & SR\ u \equiv (\exists t)(TS\ t \wedge P\ ut) \\ D_{STR} & STR\ u \equiv (\exists t_1)(\exists t_2)(TS\ t_1 \wedge TS\ t_2 \wedge O\ ut_1 \wedge O\ ut_2 \wedge \neg O\ t_1 t_2) \\ D_{SIMU} & SIMU\ uv \equiv (\exists w)(TS\ w \wedge P\ uw \wedge P\ vw) \end{array}$$

On the intended interpretation *SR* u means: Spatial regions u are parts of spacetime which, on a given $\mathcal{T}$-slicing σ are sub-regions of some time slice induced by σ. On the slicing σ the region u is not extended at all in time. By contrast, on a given slicing, spatio-temporal regions extend across time slices. This interpretation reflects at the level of the formal models that which regions of $\mathcal{ST}$ count as spatial regions depends on the underlying slicing σ.

$$\begin{array}{l} \mathsf{V}(SR) = \{\langle \mathsf{u}, \langle \gamma, \sigma \rangle\rangle \in \mathcal{D}_{ST} \times \mathcal{K} \mid \exists i \in \Re : \mathsf{u} \sqsubseteq \sigma_i(\mathcal{T})\} \\ \mathsf{V}(STR) = \{\langle \mathsf{u}, \langle \gamma, \sigma \rangle\rangle \in \mathcal{D}_{ST} \times \mathcal{K} \mid \mathsf{u} \sqsubseteq \gamma \wedge \\ \qquad\qquad \exists i, j \in \Re :\ i \neq j \wedge \mathsf{u} \cap \sigma_i(\mathcal{T}) \neq \emptyset \wedge \mathsf{u} \cap \sigma_j(\mathcal{T}) \neq \emptyset\} \\ \mathsf{V}(SIMU) = \{\langle \mathsf{u}, \mathsf{v}, \langle \gamma, \sigma \rangle\rangle \in \mathcal{D}_{ST} \times \mathcal{D}_{ST} \times \mathcal{K} \mid \exists t \in \Re : \mathsf{u} \sqsubseteq \sigma_t(\mathcal{T}) \wedge \mathsf{v} \sqsubseteq \sigma_t(\mathcal{T})\} \end{array} \tag{9}$$

The following mereological axioms for *TS* are added: distinct time slices do not overlap (A9); there are at least two non-overlapping time slices (A10); every region overlaps some time-slice (A11). If $\mathcal{ST}$ has the global or local structure of a Minkowski spacetime then there are many slicings, i.e., $\#\Sigma > 1$. In such spacetimes the axiom A_M holds requiring that simultaneity is not absolute.

$$\begin{array}{llll} A9 & TS\ u \wedge TS\ v \wedge O\ uv \rightarrow u = v & A11 & (\exists u)(TS\ u \wedge O\ uv) \\ A10 & (\exists u)(\exists v)(TS\ u \wedge TS\ v \wedge \neg O\ uv) & A_M & SIMU\ uv \wedge u \neq v \rightarrow \Diamond^{\Sigma} \neg SIMU\ uv \end{array}$$

In Minkowski spacetimes some regions of spacetime are spatial regions on some slicings but not on others. Similarly for some spatio-temporal regions.

5.2. Computational realization

In analogy to the introduction of the primitive partial ordering predicate $\sqsubseteq$ in the lattice theoretic formalization of mereology in Sec. 4, the primitive timeslice predicate of the timeslice mereology is introduced in the computational realization (i) by introducing record structures of type `('a, 'b) TS_porder_two_sort_RS_frame` as extensions of records of type `('a, 'b) porder_two_sort_RS_frame` and (ii) by introducing the locale `TS_mereology` as an extension of the locale `S5_RS_2S_GEM` (see `TS_mereology.thy`).

The first axiom of the locale `TS_mereology` (lines 5-6 of Fig. 16) mirrors the constrains of the enforced interpretation of the predicate *TS* in Eq. 8. The representation of timeslices in the computational realization (lines 5-6 of Fig. 16 and elsewhere) is somewhat simplified compared to the semi-formal presentation. In the latter slicings of spacetime are mappings of the form $\sigma : \Re \times \mathcal{T} \to \Re \times M$ (Fig. 1, Def. 1 of B.1). By contrast, in the computational representation slicings of spacetime are included explicitly as sets of the form `(s_carrier L)`. In the semi-formal presentation the set `(s_carrier L)` could be written as $\{\{u \in M \mid \exists t \in \Re : (t, u) = \sigma_t(\mathcal{T})\} \mid \sigma \in \Sigma\}$. Representing slicings as mappings is important when certain structure-preserving aspects of the mappings are used to distinguish different kinds of spacetime geometries (Sec. B.1, Sec. B.2) and the work by Bittner (2018). The focus in this paper is only on a single spacetime geometry (a simplified version of Minkowski spacetime). For this reason the mapping structure can be neglected.

The remaining axioms of the locale are respectively semantic expressions of axioms A9 – A11 of the semi-formal theory. As an example of the computational representation of the axioms in `TS_mereology.thy` the computational representation of axiom A9 is depicted in lines 8 – 9 of Fig. 16. The definitions of the predicates for spatial and spatio-temporal regions as well as the predicate of simultaneity mirror the respective specifications of the intended interpretation in the semi-formal presentation of the theory. In particular, lines 10 – 12 correspond to the first definition in Eq. 9. The original definitions of *SR*, *STR*, and *SIMU* in the object language of the semi-formal presentation will be recovered as theorems in the version of the formalized theory that is lifted to the modal level of theory presentation (Sec. 5.4).

In the context of the locale `TS_mereology` one can prove that on every slicing of spacetime the least upper bound of the set of all timeslices on that slicing is identical to spacetime itself (lines 13 – 15 of Fig. 16. Lines 13 – 14 contain the statement of the theorem and line 15 states that in the proof the (previously proved) lemmata `Set_of_TS_imp_ST` and `ST_impl_Set_of_TS` as well as the built-in proof method `blast` are used.

The locale `TS_mereology` is extended in `TS_mereology.thy` by the locale `M_TS_mereology`. The latter extends the former by adding an axiom that stipulates that simultaneity is relative (lines 16 – 20 of Fig. 16). In the context of the locale `M_TS_mereology` the axiom A_M of the semi-formal presentation will become provable at the modal level of the computational realization (Sec. 5.4).

5.3. Validity in the example model

The verification of the consistency of the axioms collected in the locales `TS_mereology` and `M_TS_mereology` and the verification their satisfaction in the example model is achieved in two steps: (1) extend the computational realization of the example model to include a structure that can serve as the interpretation for the primitive timeslice predicate; (2) verify that this model satisfies the axioms collected in the locales `TS_mereology` and `M_TS_mereology`.

Firstly, the computational realization of the example of Sec. 2.2 is extended by declaring a unary predicate `isTS_M` that holds of the time slices of a given slicing of spacetime (lines 1 – 2 of Fig. 17). To link the model to the axioms that are collected in the relevant locales a record with the name `ST_frame_M` is declared (lines 3 – 5).

The proof in which it is demonstrated that the structure `ST_frame_M` satisfies all the axioms collected in the locales `TS_mereology` and `M_TS_mereology` is realized by an exhaustive analysis of all possible cases. The resulting proof is rather lengthy and tedious and its computational realization is the theorem

```
record ('a, 'b) TS_porder_two_sort_RS_frame = "('a, 'b) porder_two_sort_RS_frame" +
   ts :: "'a => 'a RS => bool" ("TS1")

locale TS_mereology = S5_RS_2S_GEM L for L (structure) +
assumes
  "[|i ∈ r_carrier L; j ∈ s_carrier L; u ∈ carrier L|] ==>
          (TS_L u (RSC i j) = (u ∈ j))"
assumes
  "[|i ∈ r_carrier L; j ∈ s_carrier L; u ∈ carrier L; v ∈ carrier L;
          TS_L u (RSC i j); TS_L v (RSC i j); u .O_L v|] ==> u = v" and
...
 definition SR :: "_ => 'a => 'a RS => bool" ("SR1") where
   "SR_L x w ≡ (∃t. t ∈ carrier L ∧ TS_L t w ∧ x ⊑_L t) ∧ x ∈ carrier L ∧
                 r_RS w ∈ r_carrier L ∧ s_RS w ∈ s_carrier L"
...
 lemma (in TS_mereology) "[|x ∈ carrier L; i ∈ r_carrier L; j ∈ s_carrier L |] ==>
                 (x = ⨆_L{y ∈ carrier L. TS_L y (RSC i j)}) ↔ (ST_L x (RSC i j))"
   using Set_of_TS_imp_ST ST_impl_Set_of_TS by blast
...
locale M_TS_mereology = TS_mereology L for L (structure) +
assumes
  "[|SIMU_L x y (RSC i j); x ∈ carrier L; y ∈ carrier L; x ≠ y;
                 i ∈ r_carrier L; j ∈ s_carrier L|] ==>
                 (∃jj. jj ∈ s_carrier L ∧ j S_L jj ∧ ¬(SIMU_L x y (RSC i jj)))"
```

Fig. 16. Locale for time slice mereology (`TS_mereology.thy`).

`m_set_is_Inst_TS_mereology` in `ST_model_proof.thy`. It is important to acknowledge at this point that despite the tedious nature of the proof it would be rather difficult to execute a proof of this form without the computer keeping track of all the cases that must be verified.

```
abbreviation isTS_M :: "Reg => Reg RS => bool" where
   "isTS_M t w ≡ t ∈ s_RS w ∧ ((s_RS w = ts_set_M_0) ∨ (s_RS w = ts_set_M_1))"
...
abbreviation ST_frame_M where
   "ST_frame_M ≡ (| r_carrier = wl_Phys_Possible, aR = ar_WL, s_carrier = ts_set_M,
    aS = ar_TS_M, carrier = m_set, e_carrier = m_set, le = op ⊆, ts = isTS_M |)"
```

Fig. 17. Time slices and spatial regions (`ST_model_base.thy`)

5.4. *Lifting to the modal level*

Creating a modal presentation of the non-modal timeslice mereology included in the locale `M_TS_Mereology` and the associated record structures of type `('a, 'b) TS_porder_two_sort_RS_frame` is similar to creating a modal presentation of the mereology in Sec. 4.3.

Every closed modal formula in the lifted timeslice mereology is of type `('a, 'b, 'c) TS_mereology_predicate` which has two implicit arguments: (1) a record structure of type `('a, 'b, 'c) TS_porder_two_sort_RS_frame` which, as above, holds the domains of the variables, the interpretation of the axiomatic primitives, etc. and (2) the current world of type `('a RS)` (lines 1 – 2 of Fig. 18). The timeslice predicate `TS_M` of the modal language then is defined in terms of the timeslice primitive $\mathtt{TS}_\mathtt{L}$ as depicted in lines 3 – 5 of the figure. Modal versions of predicates holding of spatial regions, spatio-temporal regions, and pairs of simultaneous regions are defined analogously in terms of their non-modal counterparts (see `S5_2D_lifted_theory.thy`).

In the context of the locale `TS_mereology` one can then formally prove the all the definitions, axioms, and theorems that are stated in the semi-formal presentation of Sec. 5.1 as theorems (lines 6 – 11 of Fig 18 and `S5_2D_lifted_theory.thy`). In the context of the locale `M_TS_mereology` axiom A_M is provable (lines 12 – 13 where `Id_a_M` is a modal wrapper of the identity relation).

```
1   type_synonym ('a, 'b, 'c) TS_mereology_predicate =
2      "('a, 'b, 'c) TS_porder_two_sort_RS_frame_scheme => 'a RS => bool"

3   definition TS_M :: "'a => ('a, 'b, 'c) TS_mereology_predicate" where
4     "TS_M x L w ≡ (TS_L x w) ∧ x ∈ carrier L ∧ (r_RS w) ∈ (r_carrier L) ∧
5                   (s_RS w) ∈ s_carrier L"
    ...
6   context TS_mereology begin
7     lemma "⌊□(∀ax. SR_M x ↔ (∃at. TS_M t ∧ P_M x t))⌋"
8           unfolding SR_M_def TS_M_def P_M_def SR_def by auto
    ...
9     lemma "⌊□(∀ax. ∃ay. TS_M y ∧ O_M y x)⌋"
10          unfolding TS_M_def O_M_def using TS_and_OR by fastforce
    ...
11  end

12  lemma (in M_TS_mereology) "⌊□(∀ax y. SIMU_M x y ∧ ¬(Id_a_M x y) → ◇S(¬(SIMU_M x y)))⌋"
13     unfolding SIMU_M_def Id_a_M_def using diaS_non_SIMU by (metis r_RS.simps s_RS.simps)
```

Fig. 18. Excerpts from the lifted timeslice mereology (from `S5_2D_lifted_theory.thy`).

6. Instantiation, location, and categorization

Mirroring the methodology of Sec. 4 and Sec. 5 the introduction and axiomatization of the primitives of instantiation and atomhood are discussed starting with a review of the semi-formal presentation of Bittner (2018). As above, the computational realization is presented in three steps: (1) the axioms are expressed in a non-modal second order language with explicit reference to $\mathcal{KS}$-structures using the record structures and associated locales of Isabelle/HOL; (2) A proof is provided that the axioms are satisfied in the computational realization of the example model; (3) The axioms and definitions are lifted to the first order modal level of the formal presentation.

6.1. Semi-formal presentation

A primitive ternary relation *Inst* between two entities and a region is introduced in the object language of the formal theory. *Inst* xyu is interpreted as y is *instantiated by* x at region u (or, equivalently, x *instantiates* y at region u or x is an *instance* of y at region u). On the intended interpretation: $\mathsf{V}(\mathit{Inst}) =_{df} \mathsf{InstST} \subseteq \mathcal{D}_E \times \mathcal{D}_E \times \mathcal{D}_{ST} \times \mathcal{K}$ where the set InstST is part of the underlying $\mathcal{KS}$ structure (Eq. 1). The following axioms (some are adopted from the work by Bittner and Donnelly (2006)) are included in the formal theory to constrain InstST: if x instantiates y at u then it is not physically possible that some z is an instance of x at some region v (A12); every entity instantiates or is instantiated on some physical possibility (A13); every entity is instantiated or instantiates at spatial regions or at spatio-temporal regions (A14); if x instantiates at a spatial region then on all slicings: x instantiates at spatial regions (A15); if x is instantiated at a spatio-temporal region then on all slicings: x is instantiated at spatio-temporal regions

(A16); if x instantiates y at a spatio-temporal region u then x is uniquely located (A17); if x instantiates at two simultaneous spatial regions u and v then u and v are identical (A18).

$$\begin{array}{ll}
A12\ \mathit{Inst}\ xyu \rightarrow \neg\Diamond(\exists z)(\exists v)(\mathit{Inst}\ zxv) & A15\ \mathit{Inst}\ xyu \wedge \mathit{SR}\ u \rightarrow \Box^{\Sigma}(z)(v)(\mathit{Inst}\ xzv \rightarrow \mathit{SR}\ v) \\
A13\ \Diamond(\exists y)(\exists u)(\mathit{Inst}\ xyu \vee \mathit{Inst}\ yxu) & A16\ \mathit{Inst}\ yxu \wedge \mathit{STR}\ u \rightarrow \Box^{\Sigma}(z)(v)(\mathit{Inst}\ zxv \rightarrow \mathit{STR}\ v) \\
A14\ \mathit{Inst}\ xyu \rightarrow (\mathit{SR}\ u \vee \mathit{STR}\ u) & A17\ \mathit{Inst}\ xyu \wedge \mathit{Inst}\ xzv \wedge \mathit{STR}\ u \wedge \mathit{STR}\ v \rightarrow u = v \\
 & A18\ \mathit{Inst}\ xyu \wedge \mathit{Inst}\ xzv \wedge \mathit{SR}\ u \wedge \mathit{SR}\ v \wedge \mathit{SIMU}\ uv \rightarrow u = v
\end{array}$$

In terms of the instantiation primitive one can define: Entity x is *located* at region u if and only if there exists an entity y such that x instantiates y at u or x is instantiated by y at u (D_L); Entity x exists at timeslice t iff there is a region at which x is located and that overlaps t (D_E). An entity is *persistent* iff it is not confined to a single time-slice (D_{Pe}). Entity x is a *particular* if and only if x is a persistent entity that instantiates at some region (D_{Part}). Entity x is a *universal* if and only if x is a persistent entity that is instantiated at some region (D_{Uni}). Persistent entities are distinguished into continuants and occurrents. Entity x is a *continuant* iff x is persistent and x is located at some spatial region (D_{Cont}) . By contrast, x is a *occurrent* iff x is located at some spatio-temporal region (D_{Occ}).

$$\begin{array}{ll}
D_L\ \ L\ xu \equiv (\exists y)(\mathit{Inst}\ xyu \vee \mathit{Inst}\ yxu) & D_{Part}\ \ \mathit{Part}\ x \equiv \mathit{Pe}\ x \wedge (\exists y)(\exists u)(\mathit{Inst}\ xyu) \\
D_E\ \ E\ xt \equiv \mathit{TS}\ t \wedge (\exists u)(L\ xu \wedge O\ ut) & D_{Uni}\ \ \mathit{Uni}\ x \equiv \mathit{Pe}\ x \wedge (\exists y)(\exists u)(\mathit{Inst}\ yxu) \\
D_{Pe}\ \mathit{Pe}\ x \equiv (\exists u)(\exists v)(L\ xu \wedge L\ xv \wedge \neg \mathit{SIMU}\ uv) & D_{Cont}\ \mathit{Cont}\ x \equiv \mathit{Pe}\ x \wedge (\exists u)(L\ xu \wedge \mathit{SR}\ u) \\
 & D_{Occ}\ \ \mathit{Occ}\ x \equiv (\exists u)(L\ xu \wedge \mathit{STR}\ u)
\end{array}$$

Intuitively, $L\ xu$ means: spatio-temporal entity x is *exactly located* at region u. I.e., x takes up the whole region u but does not extend beyond u. On the intended interpretation:

$$\begin{array}{lcl}
\mathsf{V}(L) & = & \{\langle \mathsf{x}, \mathsf{u}, \kappa\rangle \in \mathcal{D}_E \times \mathcal{D}_{ST} \times \mathcal{K} \mid \\
 & & \quad \exists \mathsf{y} \in \mathcal{D}_E : \langle \mathsf{x}, \mathsf{y}, \mathsf{u}, \kappa\rangle \in \mathsf{InstST} \vee \langle \mathsf{y}, \mathsf{x}, \mathsf{u}, \kappa\rangle \in \mathsf{InstST}\} \\
\mathsf{V}(E) & = & \{\langle \mathsf{x}, \mathsf{t}, \langle\gamma,\sigma\rangle\rangle \in \mathcal{D}_E \times \mathcal{D}_{ST} \times \mathcal{K} \mid \exists \tau \in \Re : \mathsf{t} = \sigma_\tau(\mathcal{T}) \wedge \\
 & & \quad \exists \mathsf{u} \in \mathcal{D}_{ST} : \langle \mathsf{x}, \mathsf{u}, \langle\gamma,\sigma\rangle\rangle \in \mathsf{V}(L) \wedge \mathsf{u} \cap \mathsf{t} \neq \emptyset\} \\
\mathsf{V}(\mathit{Pe}) & = & \{\langle \mathsf{x}, \langle\gamma,\sigma\rangle\rangle \in \mathcal{D}_E \times \mathcal{K} \mid \exists : \mathsf{u}, \mathsf{v} \in \mathcal{D}_{ST} : \\
 & & \quad \langle \mathsf{x}, \mathsf{u}, \kappa\rangle \in \mathsf{V}(L) \wedge \langle \mathsf{x}, \mathsf{v}, \kappa\rangle \in \mathsf{V}(L) \wedge \langle \mathsf{x}, \mathsf{u}, \mathsf{v}, \kappa\rangle \notin \mathsf{V}(\mathit{SIMU})\} \\
\mathsf{V}(\mathit{Part}) & = & \{\langle \mathsf{x}, \kappa\rangle \in \mathsf{V}(\mathit{Pe}) \mid \exists \mathsf{y} \in \mathcal{D}_E : \exists \mathsf{u} \in \mathcal{D}_{ST} : \langle \mathsf{x}, \mathsf{y}, \mathsf{u}, \kappa\rangle \in \mathsf{InstST}\} \\
\mathsf{V}(\mathit{Uni}) & = & \{\langle \mathsf{x}, \kappa\rangle \in \mathsf{V}(\mathit{Pe}) \mid \exists \mathsf{y} \in \mathcal{D}_E : \exists \mathsf{u} \in \mathcal{D}_{ST} : \langle \mathsf{y}, \mathsf{x}, \mathsf{u}, \kappa\rangle \in \mathsf{InstST}\} \\
\mathsf{V}(\mathit{Cont}) & = & \{\langle \mathsf{x}, \kappa\rangle \in \mathsf{V}(\mathit{Pe}) \mid \exists \mathsf{u} \in \mathcal{D}_{ST} : \langle \mathsf{x}, \mathsf{u}, \kappa\rangle \in \mathsf{V}(L)\ \wedge \langle \mathsf{u}, \kappa\rangle \in \mathsf{V}(\mathit{SR})\} \\
\mathsf{V}(\mathit{Occ}) & = & \{\langle \mathsf{x}, \kappa\rangle \in \mathsf{V}(\mathit{Pe}) \mid \exists \mathsf{u} \in \mathcal{D}_{ST} : \langle \mathsf{x}, \mathsf{u}, \kappa\rangle \in \mathsf{V}(L) \wedge \langle \mathsf{u}, \kappa\rangle \in \mathsf{V}(\mathit{STR})\}
\end{array} \tag{10}$$

Finally, an axiom is included that ensures that every persistent entity has a worldline (A19). Region u is the worldline of entity x if and only if u a spatio-temporal region that is the mereological sum of all locations at which x is located (D_{WLOf});

$$D_{WLOf}\ \mathit{WLOf}\ xu \equiv \mathit{STR}\ u \wedge u\ \mathit{Sum}\ \{v \mid L\ xv\} \qquad A19\ \mathit{Pe}\ x \rightarrow (\exists u)(\mathit{WLOf}\ xu)$$

On the intended interpretation *WLOf* is:

$$\begin{array}{lcl}
\mathsf{V}(\mathit{WLOf}) & = & \{\langle \mathsf{x}, \mathsf{u}, \langle\gamma,\sigma\rangle\rangle \in \mathcal{D}_E \times \mathcal{D}_{ST} \times \mathcal{K} \mid \langle \mathsf{u}, \langle\gamma,\sigma\rangle\rangle \in \mathsf{V}(\mathit{STR}) \wedge \\
 & & \quad \mathsf{u} = \bigsqcup\{\mathsf{v} \in \mathcal{D}_{ST} \mid \langle \mathsf{x}, \mathsf{v}, \langle\gamma,\sigma\rangle\rangle \in \mathsf{V}(L)\} \wedge \mathsf{u} \sqsubseteq \gamma\}
\end{array} \tag{11}$$

On this interpretation all entities instantiate at or along the physically possible worldlines.

The mereological structure of the subdomain of continuants is characterized by the ternary parthood relation P_c which, holds between a time slice t and two continuant particulars x and y that are instantiated respectively at regions u_1 and u_2 such that u_1 is a part of u_2 and u_2 is part of the time slice t (D_{P_c}). By contrast, the mereological structure of the subdomain of occurrents is characterized by the binary parthood relation P_o defined as: x is part of y if and only if the location of x is part of the location of y and the location of x is a spatio-temporal region (D_{P_o}).

$$\begin{aligned} &D_{P_c}\ P_c\ xyt \equiv \mathit{Cont}\ x \wedge \mathit{Cont}\ y \wedge \mathit{TS}\ t \wedge \\ &\qquad (\exists u_1)(\exists u_2)(\exists z_1)(\exists z_2)(\mathit{Inst}\ xz_1u_1 \wedge \mathit{Inst}\ yz_2u_2 \wedge P\ u_1u_2 \wedge P\ u_2t) \\ &D_{P_o}\ P_o\ xy \equiv (\exists u_1)(\exists u_2)(\exists z_1)(\exists z_2)(\mathit{Inst}\ xz_1u_1 \wedge \mathit{Inst}\ yz_2u_2 \wedge P\ u_1u_2 \wedge \mathit{STR}\ u_1) \end{aligned}$$

On the intended interpretation P_c and P_o mean:

$$\begin{array}{rcl} \mathsf{V}(P_c) & = & \{\langle \mathsf{x}_1, \mathsf{x}_2, \mathsf{t}, \kappa\rangle \in \mathcal{D}_E \times \mathcal{D}_E \times \mathcal{D}_{ST} \times \mathcal{K} \mid \langle \mathsf{t}, \kappa\rangle \in \mathsf{V}(\mathit{TS}) \wedge \\ & & \exists \mathsf{y}_1, \mathsf{y}_2 \in \mathcal{D}_E : \exists \mathsf{u}_1, \mathsf{u}_2 \in \mathcal{D}_{ST} : \mathsf{u}_1 \sqsubseteq \mathsf{u}_2 \sqsubseteq \mathsf{t} \wedge \\ & & \quad \langle \mathsf{x}_1, \mathsf{y}_1, \mathsf{u}_1, \kappa\rangle \in \mathsf{InstST} \wedge \langle \mathsf{x}_2, \mathsf{y}_2, \mathsf{u}_2, \kappa\rangle \in \mathsf{InstST}\} \\ \mathsf{V}(P_o) & = & \{\langle \mathsf{x}_1, \mathsf{x}_2, \kappa\rangle \in \mathcal{D}_E \times \mathcal{D}_E \times \mathcal{K} \mid \exists \mathsf{y}_1, \mathsf{y}_2 \in \mathcal{D}_E : \exists \mathsf{u}_1, \mathsf{u}_2 \in \mathcal{D}_{ST} : \\ & & \langle \mathsf{x}_1, \mathsf{y}_1, \mathsf{u}_1, \kappa\rangle \in \mathsf{InstST} \wedge \langle \mathsf{x}_2, \mathsf{y}_2, \mathsf{u}_2, \kappa\rangle \in \mathsf{InstST} \wedge \mathsf{u}_1 \sqsubseteq \mathsf{u}_2, \\ & & \langle \mathsf{u}_1, \kappa\rangle \in \mathsf{V}(\mathit{STR}) \wedge \langle \mathsf{u}_2, \kappa\rangle \in \mathsf{V}(\mathit{STR})\} \end{array} \tag{12}$$

In Minkowski spacetime the parthood relation among continuants (P_c) is logically linked to the underlying slicing of spacetime. This is an immediate consequence of axiom (A_M). Only continuants that exist simultaneously at a time can be parts at that time.

The final primitive of the formal theory is the unary predicate At_e which, on the intended interpretation in $\mathcal{KS}$-structures, holds of atomic entities ($\mathsf{V}(At_e) = \mathsf{AtE} \subset \mathcal{D}_E \times \mathcal{K}$) such that the following axioms hold (in accordance with the conception of atoms in classical mechanics): There exist finitely many atomic entities (A20). If x is an atomic entity then x is an atomic entity on all physical possibilities (A21); Atomic entities are instantiated at all physical possibilities (A22); Atomic entities are instantiated at parts of time slices (A23). For every atomic entity x there is some slicing such that x is always instantiated at proper parts of time slices (A24).[3] Every atomic entity is instantiated at some non-simultaneous regions on all slicings of spacetime (A25). Distinct atomic entities do not instantiate at regions where one region is part of the other (A26).

$$\begin{array}{ll} & A22\ At_e\ x \rightarrow \Box(\exists y)(\exists u)\mathit{Inst}\ xyu \\ & A23\ At_e\ x \wedge \mathit{Inst}\ xyu \rightarrow (\exists t)(\mathit{TS}\ t \wedge P\ ut) \\ A20 \text{ finite } \{x \mid At_e\ x\} & A24\ At_e\ x \rightarrow \Diamond^{\Sigma}(t)(\mathit{TS}\ t \rightarrow (\exists u)(\exists y)(\mathit{Inst}\ xyu \wedge \mathit{PP}\ ut)) \\ A21\ At_e\ x \rightarrow \Box At_e\ x & A25\ At_e\ x \rightarrow \Box^{\Sigma}(\exists y)(\exists z)(\exists u)(\exists v)(\mathit{Inst}\ xyu \wedge \mathit{Inst}\ xzv \wedge \neg \mathit{SIMU}\ uv) \\ & A26\ At_e\ x_1 \wedge At_e x_2 \wedge \mathit{Inst}\ x_1y_1u_1 \wedge \mathit{Inst}\ x_2y_2u_2 \wedge P\ u_1u_2 \rightarrow x_1 = x_2 \end{array}$$

These axioms ensure that atoms cannot fail to be atoms and to instantiate on every physical possibility.

6.2. Computational realization of the axiomatic system

The first step of the computational realization is the extension of records of type `('a, 'b) TS_porder_two_sort_RS_frame` to include the enforced interpretations of the two new primitive predicates *Inst* and At_e in the computational representation of $\mathcal{KS}$-structures. This results in the declaration of records of type `('a, 'b) AtE_Inst_TS_porder_two_sort_RS_frame` with the slots `inst :: "'b => 'b => 'a => 'a RS => bool"` and `ate :: "'b => 'a RS => bool"` which, respectively serve as interpretations of the axiomatic primitives *Inst* and At_e. This is displayed in lines 1 – 4 of Fig. 19.

The axioms A12 – A19 of the semi-formal presentation are collected in the locale `Inst_TS_mereology`. In particular the computational representation of axiom A11 is displayed in lines 6 – 8 of Fig. 19. Similarly, the axioms A20 – A26 are collected in the locale `AtE_Inst_TS_mereology`. As an illustration the computational representation of axiom A20 is displayed in the figure. In both locales the axioms are stated

[3] In non-finite spacetimes this can be demanded for all slicings.

```
1   record ('a, 'b) AtE_Inst_TS_porder_two_sort_RS_frame =
2                                "('a, 'b) TS_porder_two_sort_RS_frame" +
3      inst :: "'b => 'b => 'a => 'a RS => bool" ("Inst1")
4      ate :: "'b => 'a RS => bool" ("AtE1")

5   locale Inst_TS_mereology = TS_mereology L for L (structure) +
6   assumes "[|(InstL x y u (RSC i j)); i RL ii; j SL jj; x ∈ e_carrier L; y ∈ e_carrier L;
7     yy ∈ e_carrier L; u ∈ carrier L; uu ∈ carrier L; i ∈ r_carrier L; ii ∈ r_carrier L;
8     j ∈ s_carrier L; jj ∈ s_carrier L|] ==> ¬(InstL yy x uu (RSC ii jj))"
    ...
9   locale AtE_Inst_TS_mereology = Inst_TS_mereology L for L (structure) +
10  assumes
11    "[|AtEL x (RSC i j); i RL ii; j SL jj; x ∈ e_carrier L; i ∈ r_carrier L;
12    j ∈ s_carrier L; ii ∈ r_carrier L; jj ∈ s_carrier L|] ==> (AtEL x (RSC ii jj))"
    ...
13  type_synonym ('a, 'b, 'c) AtE_Inst_TS_mereology_predicate =
14     "('a, 'b, 'c) AtE_Inst_TS_porder_two_sort_RS_frame_scheme => 'a RS => bool"

15  definition Inst_M :: "'b => 'b => 'a => ('a, 'b, 'c) AtE_Inst_TS_mereology_predicate"
16    where "Inst_M x y u L w ≡ InstL x y u w ∧ x ∈ e_carrier L ∧ y ∈ e_carrier L ∧
17    u ∈ carrier L ∧ (r_RS w) ∈ (r_carrier L) ∧ (s_RS w) ∈ s_carrier L"
18  lemma (in AtE_Inst_TS_mereology)
19    "⌊□(∀bx y yy. ∀au uu. Inst_M x y u → (¬(◇(Inst_M yy x uu))))⌋" unfolding
20    Inst_M_def by (metis (no_types, lifting) Inst_box_assym_P r_RS.simps s_RS.simps)
    ...
21  definition AtE_M :: "'b => ('a, 'b, 'c) AtE_Inst_TS_mereology_predicate" where
22    "AtE_M x L w ≡ AtEL x w ∧ x ∈ e_carrier L ∧ (r_RS w ∈ r_carrier L) ∧
23    (s_RS w ∈ s_carrier L)"
24  lemma (in AtE_Inst_TS_mereology) "⌊□(∀b x. AtE_M x → □(AtE_M x))⌋"
25    unfolding AtE_M_def using AtE_imp_box_AtE by auto
```

Fig. 19. The locales `Inst_TS_mereolog` and `AtE_Inst_TS_mereology` and their presentation in the modal language. (See `Inst_TS_mereolog.thy`, `AtE_Inst_TS_mereology.thy`, and `S5_2D_lifted_theory.thy`.)

at the semantic level using a second order language. The modal counterparts of these axioms that correspond to the modal formulas in the semi-formal presentation are recovered as theorems as illustrated in lines 13 – 25 of Fig. 19. Again, this mirrors the methodology discussed in Sec. 4 and Sec. 5.

6.3. *Validity in the example model*

The verification of the consistency of the axioms collected in the locales `Inst_TS_mereology` and `AtE_Inst_TS_mereology` and the verificationof their satisfaction in the example model is achieved in two steps: (1) extend the computational realization of the example model to include the new domain of entities the structures that serve as the interpretations for the primitive instantiation and Atomicity predicates; (2) verify that this model satisfies the axioms collected in the respective locales.

A computational realization of the example model which includes the new domain of entities (`thePossibleEntities`) as well as the functions `isInst_M` and `isAtE_M` that respectively serve as the interpretations for the primitive predicates `Inst` and `AtE` is displayed in Fig. 20. This figure provides an illustration of how in Isabelle/HOL many features of functional languages (like lists, maps, filters, etc.) can be used for the construction of models. The computational representation of the example model as a $\mathcal{KS}$-structure is declared as the record `AtE_Inst_ST_frame_M` in lines 25 – 29 of Fig. 20. Lines 30 – 33 display the head of the theorem in which proof it is demonstrated that the example model represented as the record

`AtE_Inst_ST_frame_M` satisfies all the axioms that are collected in the locale `AtE_Inst_TS_mereology` and its parent locales. The proof as a whole is in the file `ST_model_proof.thy`.

7. Conclusion

The aim of this paper was to demonstrate that for the development, the presentation, and the computer-assisted verification of formal ontologies the usage of higher-order languages and associated proof assistant tools are highly beneficial. For this purpose the computational realization of a semi-formal ontology that was developed elsewhere (Bittner, 2018) was employed as a case study. As formal tool for the computational realization the Isabelle/HOL/Isar framework was used. It was shown that the expressive power of the higher order logic in conjunction with a well developed infrastructure for theory and proof development facilitate (a) the formal verification of the satisfaction of the axioms of a formal ontology in a class of structures that includes the intended interpretation of the ontology and (b) the formal verification that all the theorems of the formal ontology are derivable from the axioms of the theory.

Consider Table 2 which gives an overview of the structure of the 'code' produced in the course of this case study. The logic is hierarchical and has three corresponding tiers: (i) the tier of structures that can be instantiated by specific models; (ii) the tier of axiomatization constraining classes of structures; and (iii) the tier of typed formulas that facilitate the concise presentation of the ontology. The table also illustrates that this methodology supports the separation and integration of the three levels of ontology development – (I) axiomatization; (II) model instantiation; and (III) theory presentation.

The advantages of using a framework such as Isabelle/HOL/Isar for the development of formal ontologies mirror the advantages of using modern object-orientated programming languages and associated integrated environments for software development. Locales in Isabelle/HOL like object orientation provide means for encapsulation and modularization. Developing formal ontologies in the language Isar is very similar to software development in an interpretative environment. There is a body of 'code' constituting the ontology. This body of 'code' may be distributed over various documents which dependencies are maintained by the system in the same way a compiler/interpreter maintains code dependencies. Like a compiler/interpreter Isabelle/HOL/Isar enforces the syntactic well-formedness and well-typedness of expressions. It keeps track of proof obligations. In summary, using a tool like Isabelle/HOL/Isar for ontology development feels very much feels like doing software development in an object-oriented environment – only at a higher level of abstraction. The automatically generated presentation of the fully formalized and computationally verified formal ontology can be found at: `http://www.buffalo.edu/~bittner3/Theories/OntologyCM/`.

There is a downside to the use of tools such as Isabelle/HOL/Isar. The efficient use of those tools presupposes ontologists that are highly trained and specialized in computer-assisted theorem proving (Foster et al., 2015). That is, in the same sense in which professional software development requires highly trained and specialized programmers, professional ontology development requires highly trained and specialized engineers for computer assisted theorem proving – *proof engineers*.

To illustrate the need for proof engineers, consider the example model of Sec. 2.2. Clearly, this models is overly simplistic and merely intended to illustrate two points: Firstly, how to specify a structure that can serve as an enforced model of a formal ontology and, secondly, how to use this structure in proofs that demonstrate that the axioms of the formal theory are satisfied in those structures. In contrast to the simplistic nature of the model, proofs that are collected in the file `ST_model_proof.thy` are rather lengthy, unstructured and overly complicated. They clearly do not meet the standards of an efficient and well-engineered proof. In the context of an academic paper, which aim it is to propose and to illustrate a methodology, this is not a problem. In fact, it is a good example that illustrates the need for well-trained proof engineers for *crafting* computer-verified proofs.

To illustrate this point a bit more, suppose the development and computational verification of an ontology like the one discussed in this paper in a professional (i.e., non-academic) environment. In a professional setting the lowest bar for an acceptable model for such an ontology may be a computational realiza-

```
datatype tId = Co | Oc | UC | UO
datatype eId = ZeroE | OneE | TwoE
datatype entityType = Entity tId eId "eId set"

abbreviation At_0 :: entityType where "At_0 ≡ (Entity Co ZeroE {})"
abbreviation Compl_0 :: entityType where "Compl_0 ≡ (Entity Co TwoE {ZeroE,OneE})"
abbreviation Oc_0 :: entityType where "Oc_0 ≡ (Entity Oc ZeroE {})"
abbreviation UC_0 :: entityType where "UC_0 ≡ (Entity UC ZeroE {ZeroE,OneE,TwoE})"
...
abbreviation thePossibleEntities :: "entityType set" where
  "thePossibleEntities ≡ {At_0,At_1,Compl_0,Oc_0,Oc_1,Oc_2,UO_0,UC_0}

datatype instRec = InstRec entityType entityType Reg "Reg set" "Reg set"

abbreviation instDB_M :: "instRec list" where
  "instDB_M ≡ [InstRec Compl_0 UC_0 ts0 wlCompl_0 ts_set_M_0,
               InstRec Compl_0 UC_0 ts1 wlCompl_0 ts_set_M_0,
               ...,
               InstRec At_0 UC_0 A_00 wlCompl_0 ts_set_M_0,
               ...,
               InstRec Compl_0 UC_0 ts3_M wlCompl_0 ts_set_M_1]"

definition isInst_M :: "entityType => entityType => Reg => Reg RS => bool" where
  "isInst_M e1 e2 u w ≡ (InstRec e1 e2 u (r_RS w) (s_RS w)) ∈ set instDB_M"

primrec e1_i_j_eq :: "entityType => Reg set => Reg set => instRec => bool" where
  "e1_i_j_eq ee ii jj (InstRec e1 e2 u i j) = ((ee = e1) ∧ (ii = i) ∧ (jj = j))"

definition isAtE_M :: "entityType => Reg RS => bool" where
  "isAtE_M e1 w ≡ (e1 ∈ {At_0,At_1}) ∧
                  (filter (e1_i_j_eq e1 (r_RS w) (s_RS w)) instDB_M) ≠ []"

abbreviation AtE_Inst_ST_frame_M where
  "AtE_Inst_ST_frame_M ≡ (| r_carrier = wl_Phys_Possible, aR = ar_WL,
        s_carrier = ts_set_M, aS = ar_TS_M, carrier = m_set,
        e_carrier = thePossibleEntities, le = op ⊆, ts = isTS_M, 2
        inst = isInst_M, ate = isAtE_M |)"

theorem (in AtE_Inst_TS_mereology) m_set_is_AtE_Inst_TS_mereology:
    "AtE_Inst_TS_mereology AtE_Inst_ST_frame_M"
    proof (rule AtE_Inst_TS_mereology.intro)
...
qed
```

Fig. 20. Illustration of the computational representation of entities and their instantiation according to the example model, the instantiation of a $\mathcal{KS}$-structure, and the head of the proof in which it is demonstrated that all the axioms are satisfied in the computational representation of the example model.

record type	locale	formula type	level of modal theory
RS_frame	S5_RS_frame	RS_predicate	propositional
two_sort_RS_frame	two_sort_S5_RS_frame	two_sort_RS_predicate	predicate
porder _two_sort_RS_frame	S5_RS_2S_partial_order complete_upper_semilattice partial_lower_semilattice partial_lattice S5_RS_2S_GEM	M_porder_predicate	spacetime mereology
TS_porder _two_sort_RS_frame	TS_mereology M_TS_mereology	TS_mereology _predicate	timeslice mereology
Inst_TS_porder _two_sort_RS_frame	Inst_TS_mereology	Inst_TS_mereology _predicate	instantiation and location
AtE_Inst_TS_porder _two_sort_RS_frame	AtE_Inst_TS_mereology	AtE_Inst_TS_mereology _predicate	atomic entities

Table 2

Declaration hierarchies and correspondences record types, locales, and formula types.

tion of the structures described in appendix B.1 with Def. 1 at its core. For the computational realization of such a model the proof engineer would have to have extensive proficiency in the set theory implemented as part of Isabelle/HOL and all the mathematical libraries that extend it (Paulson, 1995, 1994). In addition there are also independent archives of proof libraries (Arc, 2005) that complement the Isabelle/HOL core system. Again, this illustrates the need for well-trained and highly specialized proof engineers.

Acknowledgements

The extensive and helpful comments of the reviewers, particularly Till Mossakowski, are gratefully acknowledged. All remaining errors are solely the responsibility of the author.

References

(2005). Archive of formal proofs. http://isa-afp.org/.

Alexandroff, P. (1961). *Elementary Concepts of Topology*. Dover Publications, New York, NY.

Arfken, G. B., Weber, H. J., and Harris, F. E. (2005). *Mathematical Methods for Physicists, Sixth Edition: A Comprehensive Guide*. Academic Press, 6 edition.

Arnold, V. I. (1997). *Mathematical Methods of Classical Mechanics*. Springer.

Ballarin, C. (2004). Locales and Locale Expressions in Isabelle/Isar. In Berardi, S., Coppo, M., and Damiani, F., editors, *Types for Proofs and Programs: International Workshop, TYPES 2003, Torino, Italy, April 30 - May 4, 2003, Revised Selected Papers*, pages 34–50. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ballarin, C. (2017). HOL/Algebra/Lattice.thy.

Bateman, J. A., Borgo, S., Lüttich, K., Masolo, C., and Mossakowski, T. (2007). Ontological Modularity and Spatial Diversity. *Spatial Cognition and Computation*, 7(1).

Belot, G. (2007). The Representation of Time in Classical Mechanics. In Butterfield, J. and Earman, J., editors, *Philosophy of Physics*. Elsevier.

Benzmüller, C. (2015). HOL provers for first-order modal logics — experiments. In Benzmüller, C. and Otten, J., editors, *ARQNL 2014. Automated Reasoning in Quantified Non-Classical Logics*, volume 33 of *EPiC Series in Computing*, pages 37–41. EasyChair.

Benzmüller, C. and Woltzenlogel Paleo, B. (2015). Higher-order modal logics: Automation and applications. In Paschke, A. and Faber, W., editors, *Reasoning Web 2015*, number 9203 in LNCS, pages 32–74, Berlin, Germany. Springer. (Invited paper).

Bittner, T. (2018). Formal ontology of space, time, and physical entities in classical mechanics. *Applied Ontology*, 13(2):135–179.

Bittner, T. and Donnelly, M. (2006). A classification of spatio-temporal entities based on their location in space-time. In Meersman, R., Tari, Z., and Herrero, P., editors, *OTM 2006 Workshop Proceedings: International Workshop on Semantic-based Geographical Information Systems*, volume 4278 of *Lecture Notes in Computer Science*, pages 1626–1635. Springer-Verlag Berlin Heidelberg.

Bittner, T. and Donnelly, M. (2007). Logical properties of foundational relations in bio-ontologies. *Artificial Intelligence in Medicine*, 39:197–216.

Blanchette, J. C. (2017). Picking Nits A User's Guide to Nitpick for Isabelle/HOL Picking Nits, A User's Guide to Nitpick for Isabelle/HOL. Technical report, Institut für Informatik, Technische Universität München.

Butterfield, J. (2007). On Symplectic Reduction in Classical Mechanics. In Butterfield, J. and Earman, J., editors, *Philosophy of Physics*. Elsevier.

Champollion, L. and Krifka, M. (2015). Mereology. In Dekker, P. and Aloni, M., editors, *Cambridge Handbook of Formal Semantics*. Cambridge University Press.

Church, A. (1940). A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5(2):56–68.

Church, A. (1941). *The Calculi of Lambda-Conversation*. Princeton University Press, Princeton, NY.

Copi, I. (1979). *Symbolic Logic*. Prentice Hall, Upper Saddle River, NJ 07458.

Einstein, A. (1951). *Relativity: The Special and the General Theory*. New York: Crown Publishers Inc.

Foster, S., Zeyda, F., and Woodcock, J. (2015). Isabelle/UTP: A Mechanised Theory Engineering Framework. In Naumann, D., editor, *Unifying Theories of Programming*, pages 21–41, Cham. Springer International Publishing.

Gabbay, D. M. (2003). *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier North Holland.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2003). Sweetening Ontologies with DOLCE. *AI Magazine*, 23(3):13–24.

Haarslev, V. and Möller, R. (2003). Racer: A Core Inference Engine for the Semantic Web. In *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), located at the 2nd International Semantic Web Conference ISWC 2003, Sanibel Island, Florida, USA, October 20*, pages 27–36.

Horrocks, I. (1998). The FaCT system. In de Swart, H., editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, volume 1397 of *Lecture Notes in Artificial Intelligenc*, pages 307–312. Springer-Verlag.

Hughes, G. and Cresswell, M. (2004). *A new Introduction to Modal Logic*. Routledge, London and New York.

Jones, M. P. (1993). A system of constructor classes: overloading and implicit higher-order polymorphism. In *FPCA '93: Conference on Functional Programming and Computer Architecture, Copenhagen, Denmark*, pages 52–61, New York, N.Y. ACM Press.

Jones, S. P. and Jones, M. (1997). Type classes: an exploration of the design space. Technical report.

Kammüller, F., Wenzel, M., and Paulson, L. C. (1999). Locales A Sectioning Concept for Isabelle. In Bertot, Y., Dowek, G., Théry, L., Hirschowitz, A., and Paulin, C., editors, *Theorem Proving in Higher Order Logics: 12th International Conference, TPHOLs' 99 Nice, France, September 14–17, 1999 Proceedings*, pages 149–165. Springer Berlin Heidelberg, Berlin, Heidelberg.

Knublauch, H., Fergerson, R. W., Noy, N. F., and Musen, M. A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In McIlraith, S. A., Plexousakis, D., and Harmelen, F. v., editors, *Proc. Third International Semantic Web Conference*. Springer Verlag, Berlin.

Krifka, M. (1998). The origins of telicity. In Rothstein, S., editor, *Events and Grammar*, pages 197–235. Springer Netherlands, Dordrecht.

Lemon, O. and Pratt, I. (1997). Spatial Logic and the Complexity of Diagrammatic Reasoning. *Machine Graphics and Vision*.

Loui, M. C. (1996). Computational Complexity Theory. *ACM Computing Surveys*, 28(1).

Lowe, E. J. (2002). *A survey of Metaphysics*. Oxford University Press.

Milner, R., Tofte, M., and Harper, R. (1990). *The Definition of Standard ML*. MIT Press.

Minkowsk, H. (1908). Die Grundgleichungen für die elektromagnetischen Vorgänge in bewegten Körpern. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse.*, pages 53–111.

Mossakowski, T., Maeder, C., and Lüttich, K. (2007). The Heterogeneous Tool Set. In Grumberg, O. and Huth, M., editors, *TACAS 2007*, volume 4424 of *Lecture Notes in Computer Science*, pages 519–522. Springer Verlag.

Nipkow, T. (2003). Structured Proofs in Isar/HOL. In Geuvers, H. and Wiedijk, F., editors, *Types for Proofs and Programs (TYPES 2002)*, volume 2646, pages 259–278.

Nipkow, T., Paulson, L. C., and Wenzel, M. (2002). *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer.

Norton, J. D. (2012). Einstein's Special Theory of Relativity and the Problems in the Electrodynamics of Moving Bodies that Led him to it. In Janssen, M. and Lehner, C., editors, *Cambridge Companion to Einstein*. Cambridge University Press.

Paulson, L. (1995). Isabelle's Object-Logics. Technical report, Computer Laboratory, University of Cambridge.

Paulson, L. and Nipkow, T. (2017). Isabelle - a generic proof assistant, Retrieved January 19, 2018 from: `http://isabelle.in.tum.de/`.

Paulson, L. C. (1994). *Isabelle: A Generic Theorem Prover*. Springer Verlag.

Renz, J. and Nebel, B. (1999). On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. *Artificial Intelligence*, 108(1-2):69–123.

Simons, P. (1987). *Parts, A Study in Ontology*. Clarendon Press, Oxford.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2).

Smith, B. (2003). Ontology: An Introduction. In Floridi, L., editor, *Blackwell Guide to the Philosophy of Computing and Information Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Oxford: Blackwell.

Smith, B. (2016). Basic Formal Ontology (BFO 2.0): Specification and users guide.
Smith, B. and Varzi, A. (2000). Fiat and Bona Fide Boundaries. *Philosophy and Phenomenological Research*, 60(2):401–420.
Tarski, A. and Givant, S. (1999). Tarski's system of geometry. *Bulletin of Symbolic Logic*, 5(2):175–214.
Thompson, S. (1999). *Haskell: The Craft of Functional Programming*. Addison-Wesley, 2 edition.
Varzi, A. (2003). Mereology. In Zalta, E. N., editor, *Stanford Encyclopedia of Philosophy*. Stanford: CSLI (internet publication).
W3C OWL Working Group (2012). OWL 2 Web Ontology Language. Technical report, http://www.w3.org/TR/owl2-overview/.
Wenzel, M. (2005). Using Axiomatic Type Classes in Isabelle. Technical report, Technical University Munich.
Wenzel, M. (2017). The Isabelle/Isar Reference Manual. Technical report, Cambridge University and Technical University Munich.
Wikimedia Commons (2013). File:slope field.png — wikimedia commons, the free media repository. [Online; accessed 21-January-2018].
Wikimedia Commons (2015a). File:image tangent-plane.svg — wikimedia commons, the free media repository. [Online; accessed 21-January-2018].
Wikimedia Commons (2015b). Two coordinate charts on a manifold. Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons.

Appendix

A. Differential geometry

Bittner (2018) uses the language of the differential geometry of manifolds to explicate some of the ontological commitments underlying classical physical theories. The presentation of this subject here must remain brief and rather selective. For details see, for example, overviews by Arnold (1997); Butterfield (2007).

A.1. Manifolds

A differentiable manifold is a topological manifold with a globally defined differential structure. A topological manifold is a topological space that is locally homeomorphic to a linear (i.e. vector) space. Formally, this local structure is given by local homeomorphisms – the charts ϕ_i, mapping open subsets U_i of M to subsets of $\Re^n$ which are n-dimensional vector spaces (Fig. 21 (left)). The (global) differential structure of a manifold is built up by combining the local linear structures, local charts, to a system of atlases that cover the whole manifold such that one can reach any chart from any other chart by means of a smooth transformation. A smooth transformation or *diffeomorphism* is an invertible map that takes smooth curves to smooth curves, where a smooth curve is a curve that has derivatives of all orders everywhere. Where distinct charts overlap they must be compatible. (Fig. 21 (left)).

At every point x of a differentiable manifold M, there is a linear space T_xM 'attached' to M at x (Fig. 21 (middle)), i.e., T_xM is the *tangent space* of M at x. For all $x \in M$, T_xM has the same dimension as the manifold M at x. In planar (non-curved) manifolds like the Euclidean space $M = \Re^n$, the vectors in the tangent space T_xM at every point $x \in M$ span the whole manifold M. That is, every point $y \in M$ can be represented using a vector $\xi \in T_xM$ such that ξ begins at x and ends at y. By contrast, in curved manifolds like the surface of a sphere $\mathcal{S} \subset \Re^3$ only points in the immediate neighborhood U_x of $x \in \mathcal{S}$ can be represented by vectors in the tangent space $T_x\mathcal{S}$ (Fig. 21 (middle)). The disjoint union of all tangent spaces T_xM of M gives rise to the *tangent bundle* TM of M, i.e., $TM = \bigcup_{x \in M} (\{x\} \times T_xM)$. A point in the tangent bundle TM is a pair (x, ξ) with $\xi \in T_xM$.

Between manifolds the sub-manifold relation $\sqsubseteq$ holds. Roughly, $M_1 \sqsubseteq M_2$ if and only if M_1 is a subset of M_2, M_1 and M_2 are manifolds, and the tangent spaces of M_1 are subspaces of the tangent spaces of M_2. Join ($\bigsqcup S$) and meet ($\bigsqcap S$) operations of a non-empty set S of manifolds are such that the result is a manifold which is, respectively, the least upper or the greatest lower bound of the members of S. The mathematics of this is rather elaborate, since the whole manifold structure – including the tangent spaces – needs to be taken into account (Butterfield, 2007).

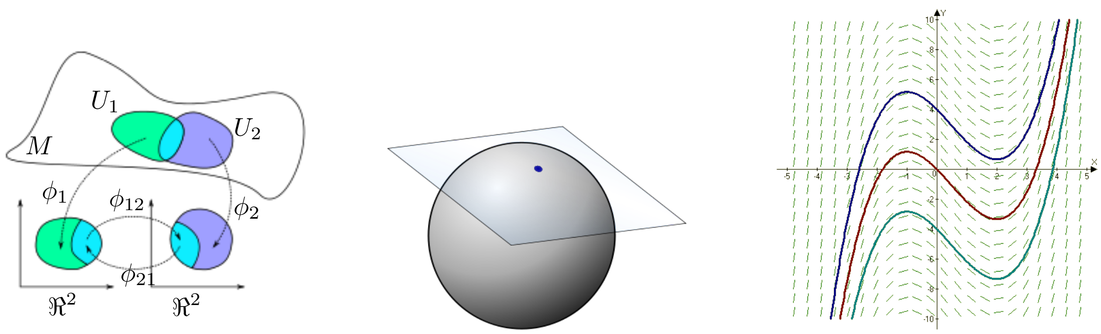


Fig. 21. Charts on a manifold (left) (Wikimedia Commons, 2015b); Tangent space $T_x\mathcal{S}$ at $x \in \mathcal{S}$ on the manifold $\mathcal{S}$ (middle) (Wikimedia Commons, 2015a). Three integral curves for the slope field corresponding to the differential equation $dy/dx = x^2 - x - 1$. (right) (Wikimedia Commons, 2013)

A.2. *Smooth curves*

A *curve* γ on a manifold M is a mapping $\gamma : \Re \rightarrow M$ from the real numbers to points of M. In what follows the letter 'γ' will be used to refer to parametric curves, i.e., functions of type $\gamma : \Re \rightarrow M$ as well as to the curves $\{\gamma(\tau) \in M \mid \tau \in \Re\}$ themselves. The context will disambiguate. If the curve γ is *smooth* then at every point $x = \gamma(\tau)$ there is a unique vector ξ in the tangent space T_xM such that ξ is tangent to γ at the point x (i.e., $\xi = \frac{d}{d\tau}\gamma(\tau)|_x$). If γ is a smooth curve on manifold M then $\gamma \sqsubseteq M$. Intuitively, the tangent space T_xM contains all possible "directions" along which a curve on M can tangentially pass through x. That is, tangent spaces arise naturally as structures formed by equivalence classes of curves on the underlying manifold.

B. Spacetime structure

B.1. *Kinematics and the spacetime structure*

The topological structure of spacetime in classical mechanics is identified with the structure of an n-dimensional Hausdorff (Alexandroff, 1961) manifold[4] with the topology $\mathcal{ST} = (\Re \times M)$ for some $(n-1)$-dimensional manifold M (Arnold, 1997; Butterfield, 2007). The topology of time is identified with the topology of the real numbers and the topology of space is identified with the topology of some Hausdorff manifold M. In classical mechanics the dimension of M is usually 3. The geometric structure of the spacetime manifold $\mathcal{ST}$ is induced by a symmetric bilinear functional $g_x : T_xM \times T_xM \rightarrow \Re$ on $\mathcal{ST}$ (Arnold, 1997) – the *metric field*. Classical mechanics includes the following postulate (illustrated in the left of Fig. 1):

Postulate 1 ((Belot, 2007; Bittner, 2018))**.** *The geometry g of the spacetime manifold $(\mathcal{ST}, g)$ singles out: (a) a distinguished class $\sigma(\mathcal{T})$ (see below) of hyper-surfaces that correspond to* instants *of time (or* time-slices*) and (b) a distinguished class Γ of curves that correspond to (geometrically) possible* worldlines *of particles.*

Let $\mathcal{ST}$ be an $n+1$ dimensional manifold with topology $(\Re \times M)$ where M is a manifold of dimension n (usually 3). In addition, let $\mathcal{T}$ be an n-(usually 3) dimensional manifold $(\mathcal{T}, g_{\mathcal{T}})$ carrying a Riemanian geometry (i.e., $g_{\mathcal{T}}$ is required to be symmetric, definite positive, and may vary smoothly) (Arnold, 1997):

Definition 1 ($\mathcal{T}$-slicing (Belot, 2007; Bittner, 2018))**.** *A $\mathcal{T}$-slicing of $(\mathcal{ST}, g)$ is a smooth map (diffeomorphism) $\sigma : \Re \times \mathcal{T} \rightarrow (\Re \times M)$ with the following properties (Illustration in the left of Fig. 1):*

[4] Roughly, in a Haunsdorff manifold there are for any distinct points $x, y \in M$ disjoint (open) neighborhoods $U_x, U_y \subset M$ such that $x \in U_x, y \in U_y$, and $U_x \cap U_y = \emptyset$.

(i) Every slice $(t, \sigma(\{t\} \times \mathcal{T})) = \{(t, \sigma_t(x)) \mid x \in \mathcal{T}\}$ of the $\mathcal{T}$-slicing σ at $t \in \Re$ is a hypersurface (an instant, a timeslice) according to the geometry g of $(\mathcal{ST}, g)$. In what follows it will be convenient to use the notation $\sigma_t(\mathcal{T})$ to refer to the timeslice $\{(t, \sigma_t(x)) \mid x \in \mathcal{T}\}$ in terms of the slicing σ;

(ii) The $\mathcal{T}$-slicing respects the worldline structure of spacetime in the sense that the set $\gamma^x =_{df} \sigma(\Re \times \{x\}) = \{(t, \sigma_t(x)) \mid t \in \Re\}$, for any $x \in \mathcal{T}$, is a possible worldline of a particle through $(t, \sigma_t(x)) \in \mathcal{ST}$ according to the geometry g of $(\mathcal{ST}, g)$, i.e., $\gamma^x \in \Gamma$.

(iii) The $\mathcal{T}$-slicing σ is such that for every $t \in \Re$ the mapping $\sigma_t : \mathcal{T} \to \sigma_t(\mathcal{T})$ is an isomorphism between $\mathcal{T}$ and $\sigma_t(\mathcal{T})$.

Def. 1 gives rise to the following naming conventions:

Definition 2. *The manifold $\mathcal{T}$ is called the* abstract instant *of the $\mathcal{T}$-slicing σ and each $\sigma_t(\mathcal{T})$ is called a* concrete *time instant of the slicing σ. The parameter $t \in \Re$ of σ_t is called coordinate time associated with σ. Σ is the set of all $\mathcal{T}$-slicing of a given underlying spacetime.*

Def. 1 is used to further constrain what is geometrically possible:

Postulate 2. *For every kinematically possible spacetime $(\mathcal{ST}, g)$ there exists a $\mathcal{T}$-slicing, i.e., $\Sigma \neq \emptyset$.*

In physical theories Postulates 1 and 2 are complemented additional kinematic and dynamic constraints that restrict what is physically possible.

B.2. Newtonian spacetime and Global Minkowski spacetime

Postulates 1 and 2 allow for a wide range of possible spacetime geometries including Newtonian spacetime and the global Minkowski spacetime of Special Relativity (Einstein, 1951; Minkowsk, 1908): Newtonian spacetime has the geometric structure of an Euclidean manifold, i.e., the geometry of $\mathcal{ST}$ is isomorphic to the geometry of $\Re^4$: $(\Re^4, \iota) \cong (\mathcal{ST}, g)$. The metric ι is a functional that is symmetric, definite positive, and the same at all points of spacetime. In such a geometry there is a *unique* slicing σ of spacetime into timeslices, i.e., $\Sigma = \{\sigma\}$. All timeslices are equipped with an Euclidean geometry that is isomorphic to the geometry of $\Re^3$. Newtonian spacetime does not place restrictions on the rate of change of location (velocity) of physically possible entities. This puts Newtonian spacetime in conflict with Classical Electrodynamics where there is a maximum for the speed of light. (Norton, 2012)

According to the theory of Special Relativity (Einstein, 1951; Minkowsk, 1908), spacetime $(\mathcal{ST}, g)$ has the structure of a manifold with topology $(\Re \times \Re^3)$ and a constant pseudo-Riemanian geometry induced by the metric η. That is, $(\mathcal{ST}, g) \cong ((\Re \times \Re^3), \eta)$. In a constant pseudo-Riemanian geometry the time-slices have an Euclidean geometry, i.e., the geometry of space is isomorphic to $\Re^3$. By contrast, *spatio-temporal* distances may be positive, zero, or negative. At every point $x \in \mathcal{ST}$ the metric $\eta(x)$ partitions spacetime in regions of positive, negative and zero distance with respect to x – the so-called light cone at x. More precisely, the metric field η of $(\mathcal{ST}, \eta)$ is symmetric and indefinite but the same at all points of spacetime. A spacetime curve γ is *time-like* if and only if the square of the length all of the tangent vectors of γ is positive[5]. The set of all time-like worldlines of a Minkowskian spacetime is:

$$\begin{aligned}\Gamma_M =_{df} \{\gamma \in \Gamma \mid \forall x \in M : \forall \tau' \in \Re : x = \gamma(\tau') \to \\ \forall \xi \in T_x M : \xi = \tfrac{d}{d\tau}\gamma(\tau)|_{\tau=\tau'} \to \eta_x(\xi, \xi) > 0\}\end{aligned} \tag{13}$$

The restriction to time-like curves in Minkowski spacetime thereby geometrically encodes the postulate of Special relativity that there is maximal velocity for particles – the speed of light.

Postulate 3. *The kinematically possible worldlines of particles in Minkowski spacetime are the time-like curves of Γ_M.*

Definition 3 (Proper time)**.** *The length of a time-like curve $\gamma \in \Gamma_M$ according to the metric η is called proper time.*

[5] Of course, the sign is pure convention which depends on the specifics of the definition of the Minkowski metric η (Minkowsk, 1908).

The topology $\mathcal{ST} = (\Re \times \Re^3)$ in conjunction with the metric η does not fix a unique $\mathcal{T}$-slicing σ of spacetime. That is, there are many distinct $\mathcal{T}$-slicings of $\mathcal{ST}$ in Σ. Proper time (Def. 3) is considered more fundamental than coordinate time (Def. 2) since it is directly linked to the underlying spacetime geometry and does not depend on a particular slicing of spacetime. This is illustrated in the left of Fig. 1.

C. Dynamics and physical possibilities

A *scalar field* $H : M \to \Re$ on a manifold M is a smooth mapping from M to the domain of scalars (real numbers $\Re$ for measurable qualities). A *vector field* $X : M \to TM$ on a manifold M is a smooth mapping from M into the tangent bundle TM so that every point $x \in M$ maps to exactly one vector $\xi \in T_xM$ of the tangent space T_xM (Fig. 21 (right)). There is a close relationship between the smooth curves of a manifold and the vector fields on that manifold. The smooth parametric curve $\gamma_{X,x} : \Re \to M$ is the *integral curve* of the vector field $X \in \mathcal{X}(M)$ through the point $x \in M$ if and only if for all $\tau \in \Re$:

$$\frac{d}{d\tau}\gamma_{X,x}(\tau) = X(\gamma_{X,x}(\tau)) \text{ and } \gamma_{X,x}(0) = x. \tag{14}$$

That is, at all points $y = \gamma_{X,x}(\tau)$ the tangent to the curve $\gamma_{X,x}(\tau)$ at y is the vector $X(y) \in T_yM$. This is illustrated in the right of Fig. 21.

In standard presentations of classical mechanics integral curves appear as the specific solutions of the differential equations that constitute the laws of physics – the equations of motion of the underlying physical system. That is, to specify the dynamics of a physical system is to identify worldlines along which physically possible processes can occur and along which physically possible particles can evolve. The essence of the Lagrangian framework of classical mechanics is to identify the dynamically, i.e., physically, possible worldlines within the larger class of kinematically possible worldlines using a scalar field that is called *The Lagrangian* ($\mathcal{L}$) which takes the tangent vectors of a manifold to the real numbers, i.e., roughly, $\mathcal{L} : T\mathcal{ST} \to \Re$. In the presentation above the Lagrangian field is assumed to be determined empirically. How to compute the vector fields which integral curves determine the physically possible worldlines is described in any textbook on classical mechanics (Arnold, 1997; Butterfield, 2007).